

T H E S E

présentée

DEVANT L'UNIVERSITE CLAUDE BERNARD - LYON

pour obtenir

LE DIPLOME DE DOCTEUR DE SPECIALITE (3ème CYCLE)

DE MATHÉMATIQUES (INFORMATIQUE)

par

Saïd KAMRAN

ALLOCATION DE RESSOURCES DANS UN RESEAU D'ORDINATEURS
--

Soutenue le 27 Septembre 1972 devant la Commission d'Examen

MM. MALECOT

Président

KOULOUMDJIAN }
MAHL }

Examineurs

T H E S E

présentée

DEVANT L'UNIVERSITE CLAUDE BERNARD - LYON

pour obtenir

LE DIPLOME DE DOCTEUR DE SPECIALITE (3ème CYCLE)

DE MATHEMATIQUES (INFORMATIQUE)

par

Saïd KAMRAN

ALLOCATION DE RESSOURCES DANS UN RESEAU D'ORDINATEURS
--

Soutenue le 27 Septembre 1972 devant la Commission d'Examen

MM. MALECOT Président

KOULOUMDJIAN }
MAHL } Examineurs



Digitized by the Internet Archive
in 2026



https://archive.org/details/IA41552613_0015

UNIVERSITE CLAUDE BERNARD

LYON

Président : M. le Professeur J. BOIDIN

1er Vice-Président : M. R. TOURAINÉ, Maître de Conférences agrégé

2ème Vice-Président : M. P. PONCET, Maître Assistant

3ème Vice-Président : M. D. SETTELEN, Etudiant

UNITES D'ENSEIGNEMENT ET DE RECHERCHE

UNITES

DIRECTEURS

U.E.R. médicale Grange-Blanche	M. le Professeur D. GERMAIN
U.E.R. médicale Alexis Carrel	M. le Professeur C. GIROD
U.E.R. médicale LYON-Nord	M. le Professeur A. BERTOYE
U.E.R. médicale LYON-Sud-Ouest	M. le Professeur L. TOLOT
U.E.R. des Sciences Pharmaceutiques	M. M. CARRAZ, Maître de Conférences agrégé
U.E.R. des Techniques de Réadaptation	M. le Professeur P. MOUNIER-KÜHN
U.E.R. de Biologie Humaine	M. J.C. CZYBA, Maître de Conférences agrégé
U.E.R. d'Education Physique et Sportive	M. A. MILLON, Professeur E.P.S.
U.E.R. des Sciences Odontologiques	M. R. VINCENT
U.E.R. de Mathématiques	M. le Doyen J. BRACONNIER
U.E.R. de Physique	M. le Professeur M. DUFAY
U.E.R. de Chimie et Biochimie	M. le Professeur D. GAUTHERON
U.E.R. des Sciences de la Nature	M. le Professeur L. DAVID
U.E.R. de Biodynamique et Psychopédagogie	M. R. FONTANGES, Maître de Conférences
U.E.R. de Physique Nucléaire	M. le Professeur A. SARAZIN
U.E.R. de Mécanique	M. le Professeur J. MATHIEU
Observatoire	M. le Professeur J.H. BIGAY
Institut Universitaire de Technologie 1	M. le Professeur B. POUYET
Institut Universitaire de Technologie 2	M. J. GALLET

Secrétaire Général : M. P. IZAUTE

Ce travail a été effectué à l'Ecole Nationale Supérieure des Mines de Saint-Etienne, sous la direction de Monsieur Robert MAHL, professeur d'Informatique de l'Ecole qui a été à l'origine de cette recherche. Qu'il veuille trouver ici l'expression de ma profonde reconnaissance pour son soutien, ses conseils et les nombreuses et fructueuses discussions qu'il m'a accordées.

Je prie Monsieur le Professeur MALECOT d'accepter mes remerciements pour l'honneur qu'il m'a fait de présider le jury de cette thèse.

Je remercie sincèrement Monsieur KOULOUMDJIAN d'avoir accepté de participer au jury de cette thèse.

Je tiens à remercier vivement Monsieur BOUCHE de m'avoir orienté et aidé à commencer ce travail.

Enfin, je remercie Mme BONNEFOY d'avoir eu la gentillesse d'assurer la dactylographie du présent volume.

RESUME

Le problème général abordé dans cette étude est celui d'allocation de ressources dans un réseau d'ordinateurs. Pour cela, après avoir essayé de donner dans une première partie un aperçu général sur l'évolution, les objectifs et les caractéristiques des réseaux et soulevé dans une deuxième partie les problèmes posés par l'allocation de ressources dans le cas d'un ordinateur en temps partagé, on s'intéresse dans la troisième partie au problème plus particulier du blocage mutuel des processus nécessitant plusieurs ressources sur différents ordinateurs. Une généralisation de la méthode de HABERMAN est proposée dans le but d'éviter ces blocages mutuels des processus. Enfin dans une quatrième partie est étudié le problème de réservations des ressources par les utilisateurs du réseau. Une simulation de réservations de ressources a été faite sur plusieurs ensembles de demandes pour permettre d'étudier les différentes politiques d'allocation de ressources.

TABLE DES MATIERES

<u>PREMIERE PARTIE</u>	page	1
GENERALITES SUR LES RESEAUX D'ORDINATEURS	"	"
I - INTRODUCTION	"	"
1°) Réseaux de terminaux	"	2
2°) Centres multi-ordinateurs	"	"
3°) Réseau spécialisé	"	3
4°) Réseau général	"	5
II - OBJECTIFS DU RESEAU D'ORDINATEURS	"	7
III - UTILISATION DU RESEAU D'ORDINATEURS	"	9
IV - PROBLEMES POSES PAR LES RESEAUX D'ORDINATEURS	"	11
V - PROBLEMES ETUDIES	"	12
 <u>DEUXIEME PARTIE</u>	"	14
Allocation de ressources dans le cas d'ordinateurs en temps partagé	"	14
I - GENERALITES	"	"
II - OBJECTIFS DES POLITIQUES D'ALLOCATION DE RESSOURCES	"	16
III - PROBLEMES POSES PAR L'ALLOCATION DE RESSOURCES.	"	17
IV - PROBLEME DE DEADLOCK	"	18
A - Théorie de Haberman pour les ordinateurs partagés.	"	19
B - Algorithme de Holt	"	22
C - Comportement du système	"	23
D - Insuffisances de la méthode précédente	"	24
 <u>TROISIEME PARTIE</u>	"	25
Stratégie d'allocation de ressources dans un réseau d'ordinateurs.	"	25
I - GENERALITES	"	"

II - PROBLEMES D'ALLOCATION DANS UN RESEAU.	page	25
III - SECURITE VIS A VIS DES ETREINTES FATALES	"	26
A - Système distribué d'ordonnancement arbitraire.	"	27
des travaux "réseaux" connu par les noeuds		
B - Procédé d'ordonnancement partiel du réseau		
permettant aux ordinateurs noeuds du réseau	"	29
d'ignorer cet ordonnancement.		
<u>QUATRIEME PARTIE</u>	"	36
Problème de réservations des ressources par les		
Utilisateurs ; étude par simulation	"	36
I - GENERALITES	"	"
II - CRITERES D'ALLOCATION DES RESSOURCES	"	37
III - STRUCTURE DU PROGRAMME DE SIMULATION	"	"
3.1. Langage de données	"	38
3.2. Analyseur Syntaxique	"	"
3.3. Structure de programme d'allocation	"	40
IV - RESULTATS	"	43
V - APPLICATIONS	"	"
<u>CONCLUSION GENERALE</u>	"	44

PREMIERE PARTIE

GENERALITES SUR LES RESEAUX D'ORDINATEURS

I - INTRODUCTION

Avant d'étudier en détail les problèmes qui sont traités dans ce papier et qui ont été l'objet direct de notre étude, il est tout à fait nécessaire d'expliquer ce qu'on entend par "réseau d'ordinateurs" et quelles sont ses propriétés et ses caractéristiques.

On s'intéressera par la suite surtout aux réseaux dits "réseaux généraux d'ordinateurs". Cette appellation permet de les distinguer des réseaux spécialisés qui sont apparus depuis 1965 et qui présentent toutes les restrictions sévères, dues à leur spécialisation, et aux capacités généralement réduites des moyens de transmission de données. La conception d'un réseau général d'ordinateurs ne préjugerait d'aucune application particulière et permettrait la communication de processus quelconques situés dans des systèmes localisés dans des ordinateurs distants et le dialogue simultané avec un ou plusieurs utilisateurs. Ce concept de "réseau général d'ordinateur" ne s'est pas dégagé brusquement mais d'une façon graduelle et en passant par différentes étapes d'évolution. Ainsi, avant de décrire les caractéristiques d'un réseau général d'ordinateurs regardons quelles sont les étapes d'évolution qui ont été franchies pour passer d'un ordinateur solitaire au concept du réseau général d'ordinateurs.

1 - RESEAUX DE TERMINAUX :

Vers 1960 est apparu le besoin de pouvoir relier directement à un ordinateur des terminaux situés à quelques centaines de mètres, ou de kilomètres. Initialement les applications qui suscitaient de tels montages étaient très spécialisées (systèmes militaires, contrôle de processus). Par la suite la technologie ainsi mise au point a été transférée à des applications commerciales ou industrielles (prise de commandes, réservations de places d'avion). Sont apparues ensuite les techniques de multi-programmation, qui ont généralisé l'emploi simultané d'un ordinateur par des utilisateurs indépendants. Deux modes d'emploi se sont alors répandus : le temps partagé, et le télétraitement par lots.

La structure de ces réseaux est logiquement de type étoilé, c'est-à-dire que les échanges d'information ont lieu entre l'ordinateur central et un terminal particulier. Il n'y a pas, normalement, d'échange entre les terminaux eux-mêmes. Ces réseaux ne sont qu'une dispersion géographique de terminaux individuels, sur les lieux d'exploitation.

2 - CENTRES MULTI-ORDINATEURS :

Dans ce qui précède, nous avons considéré que l'ensemble des terminaux d'un réseau étoilé étaient reliés directement à un seul ordinateur central. Ceci n'est vrai que dans les systèmes simples. Très souvent, le centre de traitement est en réalité un groupe d'ordinateurs interconnectés localement, par des liaisons directes de canal à canal, ou au travers d'une mémoire commune (disques, ou tampon de mémoire ferrite). Quelques exemples typiques sont les suivants :

- Ordinateur frontal (voir fig. 1)

Un ordinateur gère les terminaux et peut effectuer certains traitements sur les données entrées ou sorties. Il communique via un canal ou une mémoire secondaire avec un ordinateur de traitement.

- tandem (voir fig.2)

Deux ordinateurs identiques, ou à peu près, sont l'un et l'autre susceptibles de prendre en charge la gestion des terminaux. Mais l'un est en attente, ou occupé à des travaux de moindre priorité, alors que l'autre est en exploitation normale. En cas d'incident, le premier reprend la gestion des terminaux, pendant que l'autre est en réparation. Cette structure est classique dans les systèmes en temps réel.

- Commutateur (voir fig. 3)

Tous les terminaux peuvent accéder à l'un quelconque d'un groupe d'ordinateurs. Ceux-ci peuvent être identiques et utilisables en pool pour équilibrer la charge de traitement, ou bien ils peuvent être différents et spécialisés dans certaines catégories d'applications. Un ordinateur de liaison avec les terminaux assure la mise en communication avec l'ordinateur approprié, et peut aussi prendre en charge certains traitements (traduction de caractères, comptabilité, échanges directs de messages entre terminaux).

3 - RESEAU SPECIALISE :

Ce sont des systèmes formés de plusieurs ordinateurs éloignés les uns des autres, reliés directement par des lignes de transmission, utilisés pour un type d'application bien caractérisé. Un type de système assez répandu est le suivant :

- Commutation de messages

Un message livré à un ordinateur est retransmis à l'un de ses voisins, et ainsi de suite de proche en proche, jusqu'à ce qu'il atteigne l'ordinateur relié au destinataire. Les liaisons sont habituellement des lignes téléphoniques et forment un réseau plus ou moins maillé. Cette structure procure une plus grande fiabilité de transmission, et une meilleure utilisation des lignes, car on peut répartir le trafic sur des itinéraires différents, selon la densité de messages et l'état des circuits (panne, coupures intermittentes).

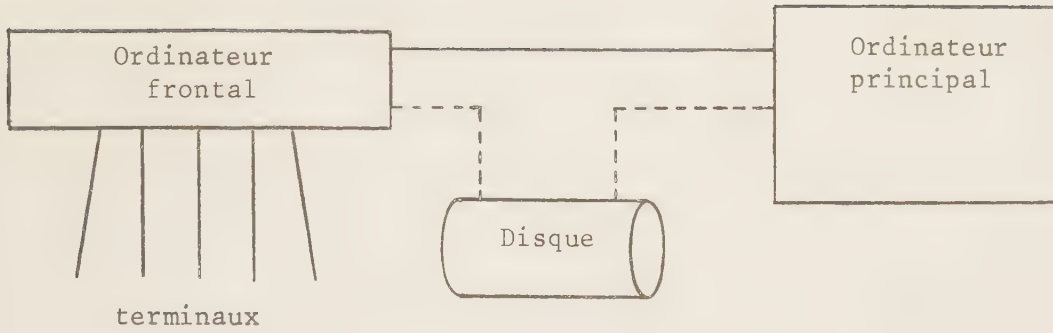


Fig. 1 - Ordinateur frontal

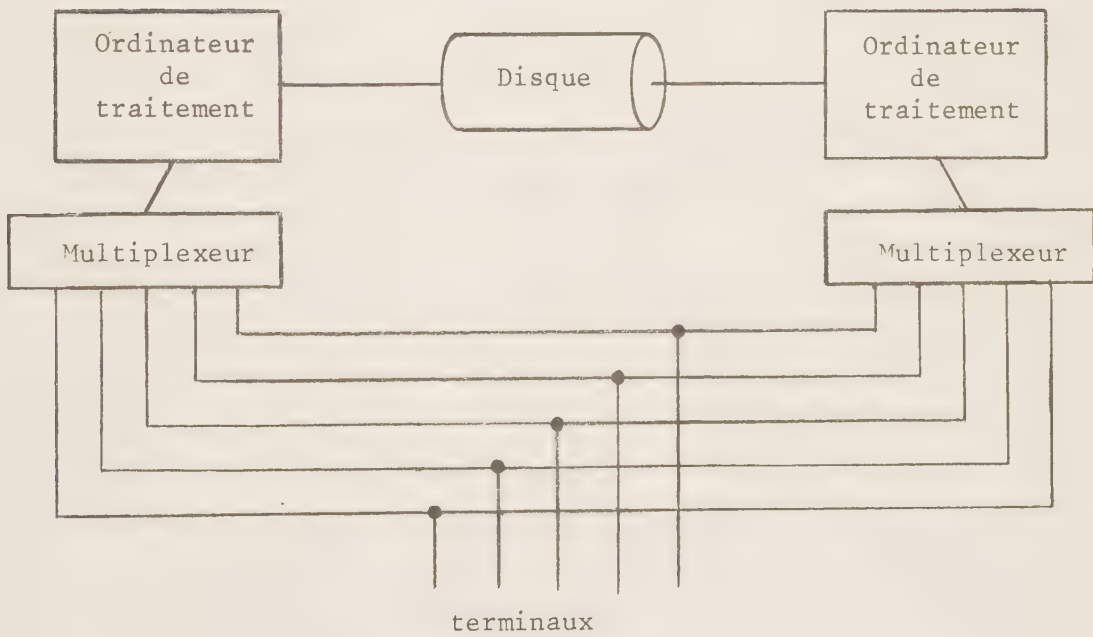


Fig. 2 - Tandem

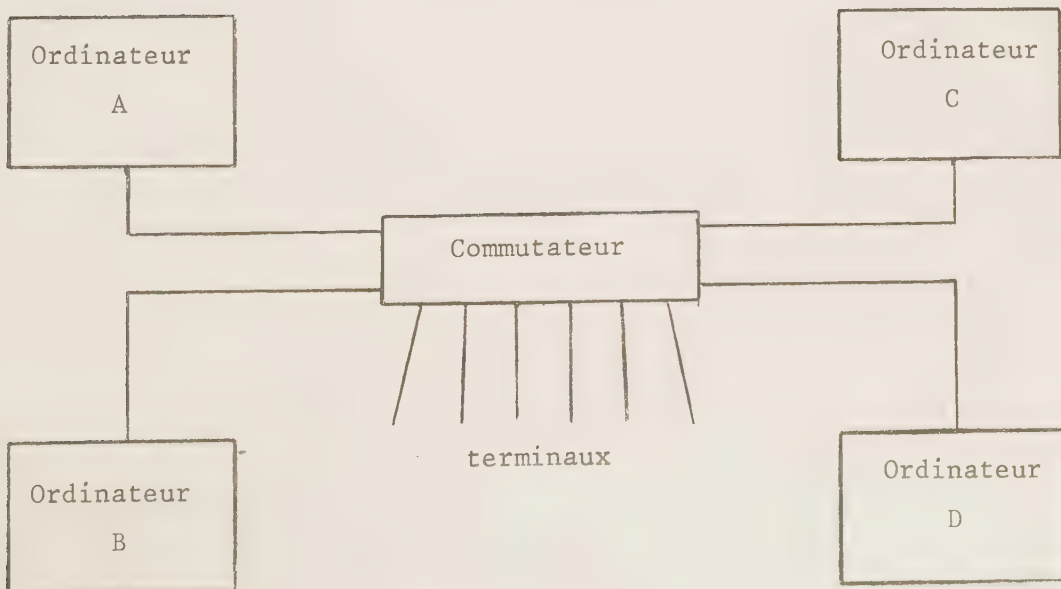


Fig. 3-Commutateur

- . Il est fréquent que les ordinateurs de commutation assurent également d'autres services, comme :
 - Traduction et édition de messages.
 - Comptabilité et historique du trafic.
 - Contrôle de validité des messages entrants.
 - Stockage et recomposition de messages à la demande.
- . Les réseaux de commutation des messages sont habituellement connectés à des ordinateurs de traitement vis à vis desquels ils agissent comme un outil de transmission de données. On est alors en présence d'un "réseau à deux niveaux" (voir fig. 5) : l'un des niveaux est composé des ordinateurs de traitement, l'autre est formé de commutateurs de messages, dont le rôle est d'abord l'acheminement des informations entre les ordinateurs de traitement et éventuellement la gestion de certains services dont nous avons donné précédemment quelques exemples.

4 - RESEAU GENERAL :

Le concept de "réseau général d'ordinateurs" s'est dégagé après celui des réseaux spécialisés, et fait actuellement l'objet de plusieurs projets expérimentaux. Il résulte de l'idée que l'on ne pourra plus concentrer indéfiniment des masses de données et des capacités de traitement de plus en plus considérables dans un petit nombre d'ordinateurs "super-puissants" tous localisés au même endroit. Cela devra faire place à de nouvelles architectures de systèmes dispersés et interconnectés. La fiabilité, la souplesse d'évolution et la gamme de services apportées par de tels réseaux devraient être sensiblement supérieures à celles d'ordinateurs individuels. La répartition de charges de travail entre différents centres permettrait d'en réduire le coût, et d'éviter les super-équipements que l'on constate fréquemment.

On étudiera dans les prochains paragraphes les objectifs, l'utilisation et les problèmes posés par un réseau général d'ordinateurs qu'on va appeler par la suite simplement le réseau d'ordinateurs.

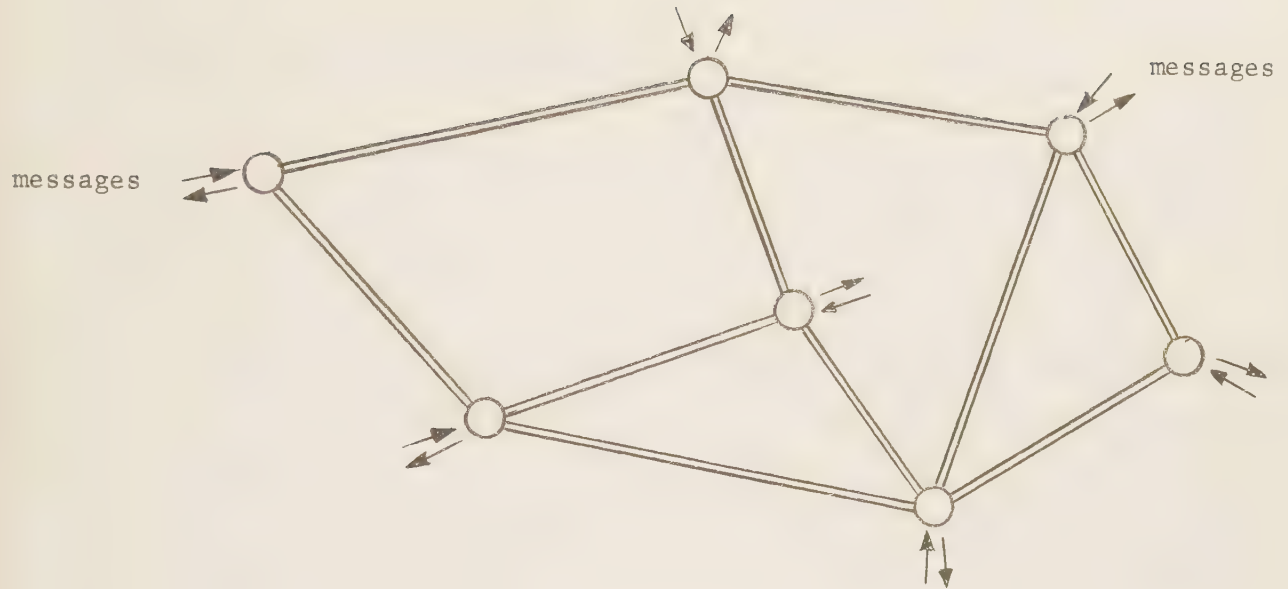


Fig. 4 - Commutation de messages

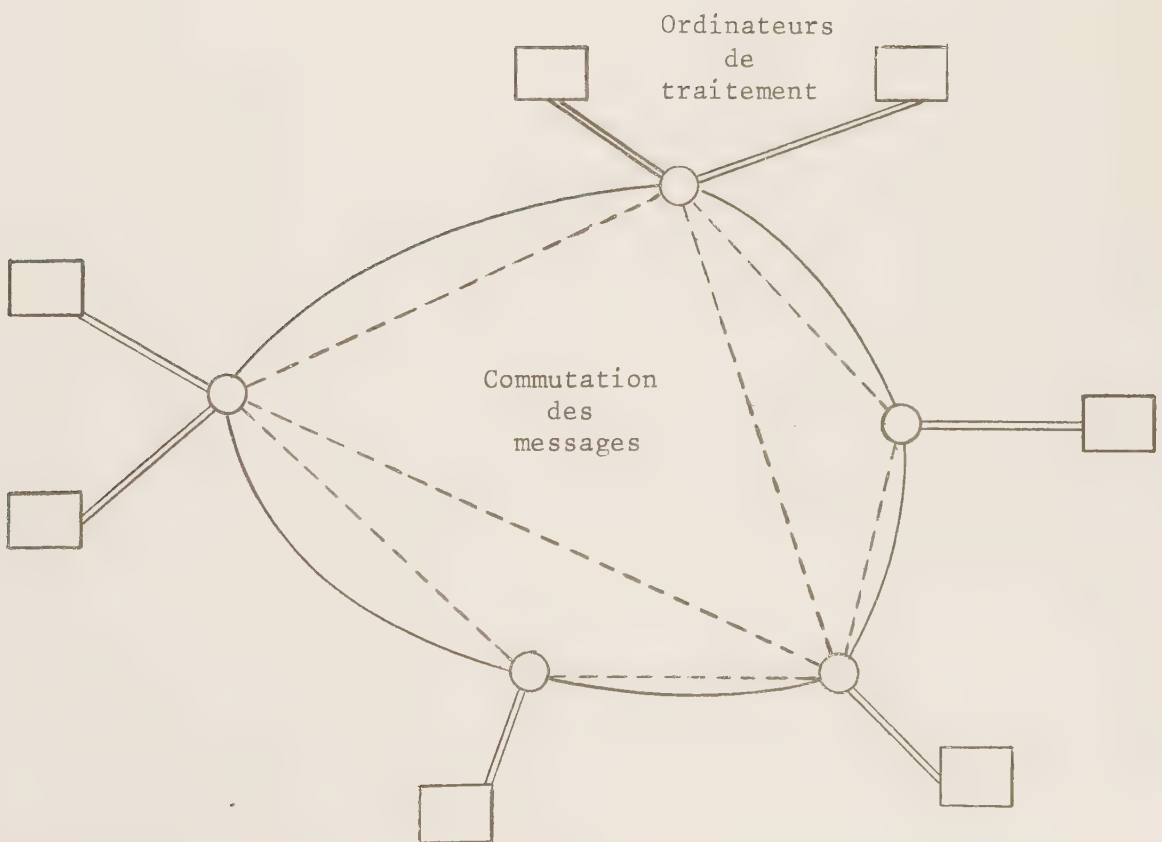


Fig. 5 - Réseau à deux niveaux

II - OBJECTIFS DU RESEAU D'ORDINATEURS

Les réseaux d'ordinateurs commencent à exister, sous des formes quelque peu expérimentales. A défaut de caractéristiques établies par l'usage, nous pouvons donner une énumération des objectifs habituellement visés :

1 - ORDINATEURS UTILISABLES POUR DES TRAVAUX A LA DEMANDE (batch à distance)

Ceci n'implique pas que tout ordinateur peut traiter n'importe quel travail, mais que dans certaines classes de service (temps partagé, traitement par lots, interrogation de fichiers, etc...) chacun des ordinateurs peut exécuter des travaux qu'il reçoit d'autres points du réseau.

2 - INTERCONNEXION EN TEMPS REEL

Les ordinateurs peuvent échanger des informations en permanence avec un temps d'accès suffisamment bref au regard des vitesses d'exécution des entrées-sorties. Ceci exclut les types de liaison dans lesquels l'information transite lentement (minutes ou heures), ou qui nécessitent un délai important pour établir la communication (ligne téléphonique commutée)

3 - TRAITEMENT MULTI-ORDINATEURS

Ceci désigne la faculté de composer des travaux dont l'exécution requiert éventuellement le concours de plusieurs ordinateurs. En d'autres termes, l'exécution mobilise des ressources de plusieurs ordinateurs : processeur central, fichiers, appareils d'entrées-sorties, mémoires secondaires, programmes spécifiques. Par exemple, une partie de travail présentant une grosse masse de calcul peut être traitée par un ordinateur rapide de type scientifique ; une deuxième partie écrite dans un langage spécialisé peut être traitée par l'ordinateur sur lequel on trouve le compilateur du langage en question ; Une troisième partie utilise des informations d'une base de données situées à un autre point du réseau.

4 - EXTENSIBILITE

De nouveaux ordinateurs peuvent être inclus dans le réseau ou retirés, sans perturber le fonctionnement général autrement que par l'apport ou le retrait de leurs propres ressources.

5 - INVULNERABILITE

Les incidents de fonctionnement d'un ordinateur n'ont pas d'autres conséquences que l'indisponibilité partielle ou totale des ressources qui lui sont attachées.

. Ces objectifs entraînent une cascade parallèle de caractéristiques d'exploitation :

- a) Les centres de traitement sont ouverts à l'extérieur mais chaque centre définira un ensemble de ressources partageables et leurs conditions d'utilisation. Le reste sera donc son domaine privé et à sa charge d'en assurer la protection contre les intrus.
- b) Les circuits utilisés pour des liaisons sont disponibles en permanence. La capacité de transmission doit être suffisante pour éviter la formation des files d'attente.
- c) L'emploi des ordinateurs du réseau est bien documenté et publié. Des procédures et des langages communs sont à définir pour les classes d'emploi les plus courantes.
- d) Les caractéristiques de raccordement et les normes d'utilisation sont conçues pour des ordinateurs hétérogènes, de fournisseurs et systèmes d'exploitation divers.

- e) Aucun ordinateur n'est indispensable au bon fonctionnement du réseau, ni susceptible d'altérer le trafic qui ne le concerne pas. Il est fondamental que les transmissions soient assurées par un système indépendant chargé de l'acheminement des messages. Ceci conduit à une structure de réseaux à deux niveaux (cf fig. 5).

II - UTILISATION DU RESEAU D'ORDINATEURS

Aucune expérience de réseau d'ordinateurs n'a encore atteint ce stade d'évolution où un utilisateur soumet un travail, non plus à un ordinateur, mais à un ensemble d'ordinateurs. Les réseaux existants n'offrent pour l'instant que des services limités à l'utilisation à distance d'un ordinateur particulier. Ainsi les limites du domaine d'utilisation des réseaux ne sont pas encore connues et un des rôles fondamentaux des projets de réseaux en cours est justement de permettre de connaître les possibilités du réseau et d'imaginer de nouvelles applications. Parmi les possibilités de l'utilisateur on peut citer :

1°) Accès aux bases de données communes :

Un effort considérable est actuellement mené dans l'administration pour constituer des bases de données communes. Dans certains cas ces informations peuvent être livrées à un client sous forme de bandes magnétiques. Plus généralement, elles ne sont pas transportables pour être exploitées sur un autre ordinateur, à cause de multiples raisons : incompatibilité de code software spécial, volume prohibitif, renseignements confidentiels. En l'absence des terminaux à distance, les clients auraient à soumettre leurs demandes par courrier, ou téléphone. Les coûts de traitement et le temps de réponse en seraient sérieusement gonflés. Les réseaux d'ordinateurs permettent un accès direct à partir de terminaux légers ou lourds, soit sous forme d'interrogation en ligne, soit sous forme de dépôt d'une requête avec traitement différé, mais automatique. Il est simplement fait appel au software d'accès

normalement prévu par le centre gestionnaire de la base de données. Toute procédure de contrôle, comptabilité ou restriction d'accès peut être prévue. En l'absence de réseau, il faudrait que chaque utilisateur puisse obtenir l'accès, via circuit loué ou commuté, à toutes les bases de données qui l'intéressent. Du fait de la simultanéité possible des requêtes, et pour éviter de trop longues attentes, l'ordinateur associé à la base devrait être muni d'autant de circuits d'échange et de matériels d'interface que de communications possibles. De plus les particularités des procédures d'échange et des terminaux rendraient dans bien des cas la communication impossible, en l'absence de normes reconnues dans ce domaine. Dans le cas d'un réseau il suffit d'un seul canal d'échange avec le réseau (ou deux) pour être mis en communication avec autant de transactions simultanées que peut en supporter le software ou la charge de traitement de l'ordinateur. Du fait de la rapidité de transmission du réseau, l'accès aux bases de données ne se résume pas seulement à des consultations ou extractions d'informations à des fins ultérieures. Lorsque les exploitations des ordinateurs connectés auront atteint une fiabilité élevée, il deviendra commode de placer les accès aux données dans le corps même des programmes qui en ont besoin. Ceci limitera les transferts massifs d'information simplement au cas où l'on en aurait besoin. En d'autres termes, un réseau d'ordinateurs permet d'évoluer vers des traitements intégrés, avec des bases de données décentralisées.

2°) Partage de programme :

Il n'y a pas de différence de nature entre un programme et un fichier de données quelconques, du point de vue du stockage ou de la transmission de données. Rien ne s'oppose donc à faire "voyager" des programmes, sauf qu'en général ils ne s'exécutent correctement que sur l'ordinateur d'origine. Moyennant des précautions bien respectées et des systèmes de même famille, on peut échanger des programmes sans adaptation manuelle ; mais c'est plutôt l'exception.

Par contre, il est en général plus facile d'utiliser un programme sur son ordinateur de résidence, car le transport des données ne pose pas habituellement de problèmes d'adaptation difficiles. Par contre, les volumes, délais d'acheminement, inaccoutumance à un centre extérieur, sont les obstacles principaux.

Avec un réseau, l'utilisateur peut faire appel aux services d'un autre ordinateur pour tout ou partie d'un traitement, c'est-à-dire activer à distance un compilateur, une recherche documentaire, un simulateur, etc... qui ne serait pas disponible sur son ordinateur habituel. Les transferts de données associés à ces exécutions peuvent être effectués à la demande, ou en début et fin de travail.

3°) Partage d'équipement :

Exécuter des programmes, rechercher des données sur un autre ordinateur est déjà une manière de partager du matériel, bien qu'elle résulte seulement du partage d'autre chose. Une manière de partage d'équipement est celle des ressources uniques, comme une imprimante à microfilm, ou une sortie graphique. La multiplicité et la spécialisation des périphériques interdisent à chaque centre de pouvoir offrir tous les services possibles. Seuls certains centres ont une charge suffisante pour justifier l'acquisition d'un équipement spécial. La mise à disposition de cet équipement à l'intérieur d'un réseau permet à tous les usagers d'en solliciter l'utilisation à distance.

- PROBLEMES POSES PAR LES RESEAUX D'ORDINATEURS

L'expérience acquise dans le domaine des réseaux est encore fragmentaire. S'il est démontré qu'un réseau peut fonctionner (ex. réseau ARPA), il est également clair que l'on n'a pas encore dépassé le stade du prototype. Autrement dit, une investigation considérable reste à faire pour que ce genre d'outil puisse être construit et utilisé à une échelle nationale et internationale dans des

conditions satisfaisantes de sécurité, fiabilité, économie et utilisation. Ainsi le nombre de problèmes relatifs au sous-système de commutation de messages ou au réseau des ordinateurs de traitement , susceptibles d'approfondissement est important. Parmi les sujets possibles de recherche on peut citer :

a) Transmissions :

- Optimisation conjointe des capacités de transmission et de stockage de l'information (sous-réseau de transmission)
- Contrôle de trafic d'un réseau de commutation de message.
- Compression d'information
- Techniques de transmission à grand débit.

b) Bases de données :

- Structure de base de données répartie sur plusieurs ordinateurs.
- Structure d'un système spécialisé dans le stockage de données.
- Conversion de bases de données entre ordinateurs hétérogènes.
- Sécurité des informations, et moyens de protection.

c) Système d'exploitation du réseau :

- Ressources virtuelles dans un réseau.
- Communications entre programmes.
- Confection de programmes répartis dans un réseau

d) Modèles et langages divers

- PROBLEMES ETUDIES

Parmi les objectifs d'un réseau d'ordinateurs cités dans le § II c'est surtout le troisième c'est-à-dire "le traitement multi-ordinateurs" qui nous a intéressé dans notre étude. Il semble d'ailleurs que c'est l'objectif qui est le

plus difficile à atteindre et soulève le plus de problèmes théoriques. Si on prend l'exemple le plus typique de réseau général déjà réalisé c'est-à-dire ARPANET, on voit que la caractéristique 3 n'a pas encore dépassé le stade de démonstration. D'autre part, la plupart des thèmes de recherche sur les réseaux évoqués au § IV rentrent dans le cadre plus général du problème de traitement multi-ordinateurs. Que signifie le traitement multi-ordinateurs ? Nous l'avons défini comme la possibilité de composer des travaux dont l'exécution requiert le concours de plusieurs ordinateurs. En d'autres termes un travail peut mobiliser pendant un certain temps un certain nombre de ressources sur différents ordinateurs du réseau. Dans les réseaux homogènes se pose le problème suivant : étant donné un travail susceptible d'être exécuté dans un ordinateur quelconque du réseau, quel ordinateur choisir pour effectuer le traitement ? Ce problème ne sera pas abordé dans la suite. Nous nous intéressons ici aux réseaux hétérogènes où un travail indique explicitement quelles ressources de quels ordinateurs il désire utiliser sans laisser au système aucun choix concernant les ordinateurs devant participer au traitement. D'où la nécessité d'assurer une synchronisation entre les différents ordinateurs exécutant un ou plusieurs travaux communs. On voit que le problème étudié est le problème très général d'allocation des ressources que l'on rencontre chaque fois qu'il y a un certain nombre de ressources et que plusieurs usagers sollicitent l'utilisation de ces ressources.

Dans la suite on s'intéresse aux problèmes d'allocation de ressources "non-préemptibles" . Deux aspects d'allocation de ressources ont été étudiés : Dans la troisième partie est étudié le problème du blocage mutuel des processus (problème de deadlock) utilisant les ressources d'un réseau d'ordinateurs ; Dans la quatrième partie on a étudié le cas où il y a la possibilité de réservation de certaines ressources pendant un certain temps par les utilisateurs du réseau.

Une ressource est préemptible (exemple : unité centrale, canal) si c'est seulement le système qui peut la faire travailler ou la libérer. Par contre une ressource non-préemptible (ou dont le coût de préemption est élevé : unité de bande magnétique, unité de disque) peut être libérée seulement par les processus qui l'utilisent.

DEUXIEME PARTIE

ALLOCATION DE RESSOURCES DANS LE CAS
D'ORDINATEURS EN TEMPS PARTAGE

- GENERALITES

Une des caractéristiques de la troisième génération d'ordinateurs est l'existence d'un mécanisme d'allocation automatique (contrôlé par le système) de ressources. Les raisons de centraliser le mécanisme d'allocation de ressources sont les suivantes :

- 1°) Le programmeur est plus productif quand il n'est pas concerné par les problèmes d'allocation de ressources qui s'ajoutent aux problèmes de la structure logique de son propre algorithme. Faciliter la tâche du programmeur est d'ailleurs le but principal des langages de programmation qui permettent au programmeur d'ignorer les détails des opérations-machine.
- 2°) Du fait de l'imprévisibilité et la concurrence des demandes, le programmeur est incapable de prendre des décisions efficaces d'allocation de ressources.
- 3°) Un allocateur central de ressources est capable de surveiller l'ensemble du système et de contrôler l'usage des ressources dans le but de satisfaire les deux objectifs qui sont : l'efficacité du système et le bon service des usagers.

La plupart des systèmes de temps partagé sont organisés de la façon suivante : à chaque instant un programme peut être dans l'un des trois "états" suivants :

1) Etat prêt :

C'est l'état d'un programme qui demande l'utilisation de la mémoire centrale et des processeurs et n'a pas obtenu encore les ressources demandées

2) Etat actif :

C'est l'état du programme qui est en mémoire centrale.

3) Etat bloqué :

C'est l'état du programme qui demande l'utilisation des équipements périphériques ou qui attend un message d'un autre programme.

La transition entre les différents états est représentée ci-dessous :

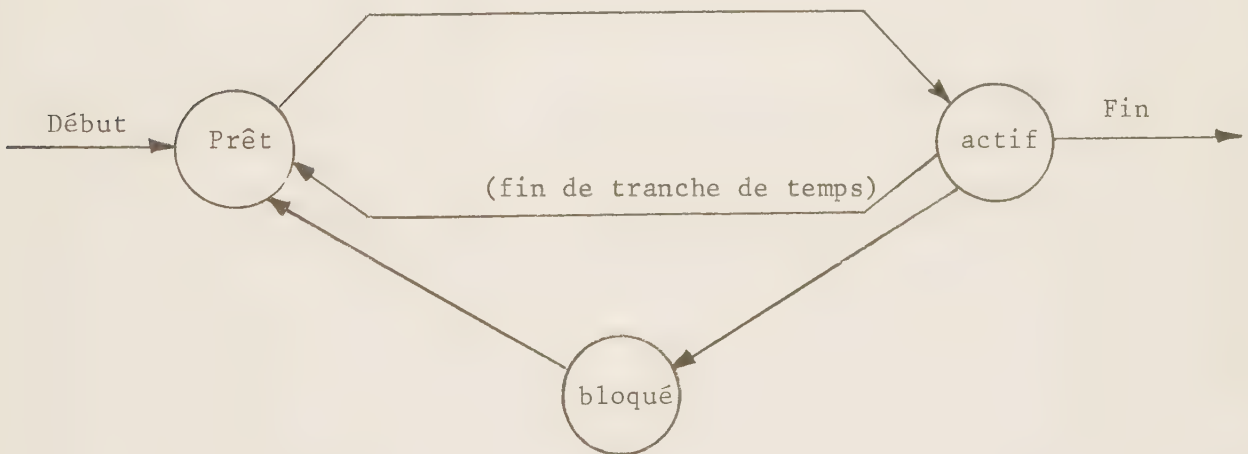


fig. 6

L'existence des états d'un programme est matérialisée par les différentes files d'attente organisées dans l'ordinateur. A chaque état correspond une file. Aussi les programmes prêts sont distribués parmi les différents "niveaux" d'une queue dont chaque niveau correspond à la demande d'utilisation d'une seule

ressource. De même les programmes actifs peuvent être divisés en deux catégories : "nouveaux" s'ils sont initialisés récemment ou "anciens" dans le cas contraire. Enfin les programmes bloqués sont regroupés suivant les raisons de leur blocage.

OBJECTIFS DES POLITIQUES D'ALLOCATION DE RESSOURCES

Les objectifs sont de deux types distincts et souvent en conflit : l'efficacité du système et la satisfaction des usagers. La figure ci-dessous illustre le conflit qui existe entre la satisfaction des usagers (temps d'attente pour obtenir une ressource) et l'efficacité du système (fraction de temps où un processeur est inutilisé) tel qu'on le rencontre dans le cas d'un système de temps partagé monoprogrammé.

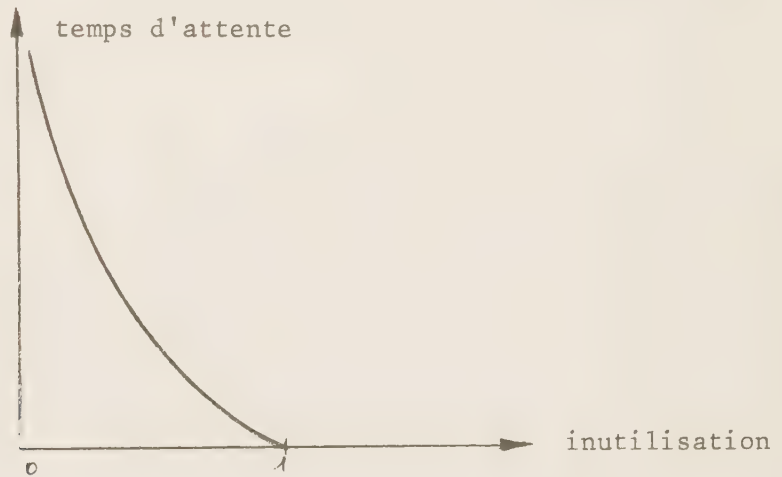


Fig. 7

Ce conflit résulte de la théorie des files d'attente qui affirme que plus les usagers qui demandent un processeur sont nombreux (donc plus le processeur est utilisé), plus la file d'attente est longue (donc plus l'utilisateur doit attendre pour être servi). Ainsi si α est le coefficient d'inutilisation d'un processeur, le temps d'attente espéré dans la queue est proportionnel à $(1 - \alpha)/\alpha$.

I - PROBLEMES POSES PAR L'ALLOCATION DE RESSOURCES.

Parmi les ressources d'un ordinateur, la mémoire joue un rôle particulier. Les techniques et les politiques de gestion de la mémoire ont attiré beaucoup d'attention et ont été l'objet de nombreuses études approfondies. En effet, c'est la ressource la plus demandée et indispensable pour la plupart des travaux utilisant d'autres types de ressources. On distingue plusieurs types de mémoires, mais dans tous les systèmes il y a au moins deux niveaux : la mémoire rapide et la mémoire auxiliaire. Etant donné la taille limitée de la mémoire rapide, il y a toujours assez d'utilisateurs en concurrence pour qu'elle soit insuffisante. Alors se pose le problème crucial : comment faire pour qu'à chaque instant figure en mémoire ce dont le programme actif a besoin, et comment réduire au minimum le nombre des mots échangés entre mémoires rapide et auxiliaire puisque ces échanges freinent l'ordinateur ? Les solutions proposées sont nombreuses et ont été l'objet de longues études. On ne va pas les étudier en détail par la suite, mais seulement les énumérer : les premières techniques consistaient à donner aux utilisateurs une place fixe en mémoire qui pouvait contenir un ou plusieurs programmes en même temps. Les techniques nouvelles ne veulent pas définir, à priori la place offerte à chaque utilisateur. Une partie de la mémoire est réservée au système, le reste est octroyé aux utilisateurs suivant leurs besoins. Cela implique le partage de l'espace disponible en mémoire en un certain nombre d'unités indivisibles. On accorde à chaque utilisateur un nombre entier de ces unités suivant ses besoins. Mais comme il est difficile d'espérer qu'en toutes circonstances on trouvera pour un programme le nombre voulu des unités consécutives, il faut chercher par quelles techniques on peut implanter en mémoire un travail dans des parties non contigües. Il y a deux manières d'aborder le problème, l'une liée à la structure logique des programmes et appelée "segmentation", l'autre liée à la structure câblée des machines et appelée "pagination". Cela a permis de distinguer l'adresse d'une donnée dans un programme et l'adresse physique effective de cette même donnée en mémoire. Ce qui a conduit à la notion de "mémoire virtuelle". A part les problèmes posés par la gestion de la mémoire, évoqués brièvement ci-dessus et qui n'ont pas été l'objet de notre étude, on peut séparer les problèmes posés par l'allocation de ressource dans le cas d'ordinateurs en temps partagé en deux classes :

A - Les problèmes d'allocation de ressources préemptibles

B - " " " " " non préemptibles

Un système d'exploitation traite généralement ces problèmes dans l'ordre suivant :

- Allocation de ressources non-préemptibles
- Allocation de ressources préemptibles aux travaux disposant des ressources non-préemptibles dont ils ont demandé l'usage.

Dans la résolution des problèmes A et B, on cherche généralement à satisfaire deux types de critères :

- a) Un critère impératif : le système ne doit pas tomber dans une impasse par suite du blocage mutuel de divers processus (deadlock)
- b) Un critère d'efficacité : le fonctionnement du système doit chercher à maximiser une certaine fonction d'efficacité (dont la définition est souvent difficile)

Le critère impératif d'éviter le deadlock ne survient que lors de l'allocation de ressources non-préemptibles (type B). Le critère d'efficacité est présent aussi bien dans les allocations de type A que dans les allocations de type B. L'allocation de ressources préemptibles est souvent désignée par micro-allocation et l'allocation de ressources non préemptibles par macro-allocation, afin de distinguer les intervalles de temps sur lesquelles elles portent. Les problèmes de micro-allocation et de macro-allocation ne peuvent d'ailleurs pas être traités indépendamment (cf.[6] pour des exemples de micro et de macro-algorithmes liés dans les ordinateurs partagés).

- PROBLEME DE DEADLOCK (EMBRASSE MORTELLE ; ETREINTE FATALE; etc...)

Le problème de deadlock apparaît dans les systèmes de traitement multi-ordinateurs et dans le cas d'allocation de ressources non-préemptibles. Dans ce cas la somme des quantités de ressources demandées par l'ensemble des processus peut dépasser la capacité totale des ressources tandis que la quantité demandée par chaque processus reste inférieure à la capacité totale sinon aucun processus

ne peut progresser. Alors une situation de deadlock apparaît si aucun processus ne peut plus continuer ses travaux parce qu'il n'arrive pas à s'approprier des ressources dont il a besoin et qui sont détenues par le voisin.

La figure 8 illustre les caractéristiques du problème de deadlock. Elle montre le cas simple de deux processus P_1 et P_2 qui, pour effectuer leurs travaux demandent l'utilisation de deux ressources R_1 et R_2 . Un processus ne peut s'approprier une ressource que si elle est complètement libérée par l'autre. Un chemin de cet espace à deux dimensions est une suite d'étapes irréversibles où chacune représente l'avancement de l'un des processus. D'après les hypothèses faites, la zone hachurée est impossible et la zone pointillée est "unsafe" puisqu'il n'y a aucune issue une fois qu'on y est entré. On peut déduire de l'exemple précédent qu'il existe trois conditions pour un risque de deadlock.

- 1°) Chaque processus peut exiger le contrôle exclusif des ressources qu'il détient
- 2°) Les ressources sont non-préemptibles c'est-à-dire qu'elles peuvent être libérées seulement par les processus qui les détiennent.
- 3°) Chaque processus qui possède un type de ressource peut demander l'utilisation d'un autre type de ressource.

Il n'est pas toujours possible d'éviter une situation de deadlock quand on se trouve dans l'un des deux premiers cas cités ci-dessus. Cependant on peut empêcher systématiquement le deadlock dans le troisième cas en employant une méthode appelée : "utilisation ordonnée de ressources". Cette méthode consiste à ordonner les ressources $R_1, R_2, \dots, R_n$ et à ne pas permettre à un processus qui possède une ressource R_i d'utiliser une autre ressource R_j que si $i < j$.

A - THEORIE DE HABERMAN POUR LES ORDINATEURS PARTAGES.

La plus grande difficulté dans le problème de deadlock est de détecter une situation de deadlock et trouver dans l'espace des avancements des travaux (voir l'exemple précédent), les régions dangereuses ou "unsafe". Pour cela il faut formaliser le problème et trouver des modèles d'attachement et de détachement des ressources aux processus.

HABERMAN [1] propose de résoudre le problème en associant à chaque processus deux vecteurs :

- Le vecteur $\vec{A}$ des ressources actuellement allouées au processus.
- Le vecteur $\vec{B}$ des ressources maximum nécessitées par le processus pour pouvoir terminer son déroulement.

Si m est le nombre de processus et n le nombre de ressources, alors a_{ij} est la quantité de ressources de type j allouées au processus i , et b_{ij} la quantité maximum de ressources j allouable à i pour lui permettre de se dérouler. La quantité r_j ($1 \leq j \leq n$) de ressources de type j dans le système est supposée connue également.

L'état est dit "safe" si aucun risque d'étreinte fatale n'existe, c'est-à-dire s'il existe un ordre $i_1, i_2, \dots, i_m$ dans lequel on peut traiter les processus en libérant toutes les ressources utilisées par un processus dès lors que celui-ci est terminé. Donc à priori, pour déterminer si une situation donnée est "safe", il faut examiner les ordres possibles de passage des processus, ce qui prend un temps proportionnel à $m!$ (si m est le nombre de processus). Or l'algorithme de Haberman nécessite un temps proportionnel à m^2 . Cet algorithme repose sur la remarque suivante : "Si une situation est safe avec m processus, elle restera safe après que le premier processus ait terminé son travail et relâché les ressources qu'il détenait c'est-à-dire avec $(m-1)$ processus et ainsi de suite". Ainsi au lieu d'examiner systématiquement tous les ordres possibles, il suffit de chercher un processus qui puisse progresser et terminer son travail. Après avoir rendu la quantité de ressources qu'il détenait à la quantité disponible de ressources on cherchera parmi les restants le premier qui peut progresser et ainsi de suite. Si à une étape donnée de notre recherche, on ne trouve aucun processus qui puisse progresser, on peut conclure que la situation donnée est unsafe sans avoir à examiner d'autres ordres possibles de passage des processus. Ainsi on voit que l'algorithme demande un temps proportionnel au carré du nombre de processus.

B - ALGORITHME DE HOLT

Cet algorithme est plus performant que celui d'HABERMAN. Il cherche si un état donné est safe en un temps proportionnel au nombre de processus. L'algorithme travaille sur plusieurs tableaux :

- 1°) Un tableau A à deux dimensions = $m \times n$ (m étant le nombre de processus et n le nombre de ressources). On note dans ce tableau pour chaque processus la quantité qu'il a besoin sur chaque type de ressource pour finir son travail. Pour chaque ressource les processus sont ordonnés (par chaînage) suivant l'ordre décroissant des quantités demandées.
- 2°) Un tableau B de mêmes dimensions ($m \times n$) qui contient 0 ou 1 suivant que pour chaque type de ressource le travail de chaque processus peut finir ou non. Ce tableau est initialisé à 0.
- 3°) Un tableau C de dimension n qui permet de garder la quantité disponible à chaque instant sur chaque ressource.
- 4°) Un tableau D de dimension m qui donne à chaque instant le nombre de ressources qui peuvent être utilisées pour finir le travail de chaque processus. Ce tableau est initialisé aussi à 0.

L'algorithme de recherche procède de la façon suivante :

- a) On considère la première colonne du tableau A c'est-à-dire la première ressource). A l'aide du tableau C et le chaînage de A on cherche le premier processus qui peut finir son travail en ce qui concerne cette ressource. On note alors la position du processus dans le tableau A (par la valeur d'un pointeur).

- b) On met à 1 la case correspondante à ce processus et cette ressource dans le tableau B
- c) On ajoute 1 au contenu de la case correspondante de ce processus dans le tableau D.
- d) On ajoute au contenu de la case correspondante à la ressource courante dans le tableau C la valeur de la case de A montrée par le pointeur.
- e) On modifie la valeur du pointeur en adressant le processus suivant donné par le chaînage et on recommence par a.

Après avoir examiné toutes les colonnes (c'est-à-dire toutes les ressources) on regarde le contenu du tableau D. L'état donné est safe si tous les éléments de D sont égaux à n (nombre de ressources).

En étudiant l'algorithme de près, on voit que le temps demandé est proportionnel à $m \times n$. Etant donné qu'en général n est beaucoup plus petit que m, le temps sera donc en gros proportionnel à m.

C - COMPORTEMENT DU SYSTEME

Le système répond aux questions de l'utilisateur comme suit :

- a) L'utilisateur demande une variation de son vecteur $\vec{A}$ de ressources allouées (avec $\vec{A} \leq \vec{B}$)

Le système détermine si cette augmentation le laisse dans l'état safe ; si c'est le cas, l'utilisateur peut recevoir son nouveau vecteur d'allocation. Sinon il doit attendre un temps fini correspondant au traitement d'autres utilisateurs qui, eux peuvent être traités de façon safe.

- b) L'utilisateur demande un certain vecteur $\vec{B}$ alors que son vecteur $\vec{A}$ d'allocations actuelles est nul.

Le système le lui alloue éventuellement (algorithme de démarrage des travaux) ; cette allocation n'est jamais en contradiction avec la "safety" mais dépend de considérations d'optimisation du choix des travaux à traiter.

- c) L'utilisateur exige un vecteur $\vec{B}$ supérieur au vecteur $\vec{B}$ actuel, alors que son vecteur $\vec{A}$ est non nul.

Si le système trouve que cette requête le rend "unsafe", il détruit le travail de cet usager, libérant toutes ses ressources ; ou bien, il peut marquer le travail comme susceptible d'être détruit.

- d) L'utilisateur demande si un vecteur $\vec{B}$ supérieur à son vecteur d'allocation maximum actuel pourrait lui être alloué, et si oui, demande que cette allocation ait lieu.

Le système lui alloue éventuellement un nouveau vecteur $\vec{B}$ ou bien continue à le faire travailler avec l'ancien $\vec{B}$.

D - INSUFFISANCES DE LA METHODE PRECEDENTE

- a) Elle n'évite pas qu'un travail puisse être délayé indéfiniment. HOLD [2] résoud ce problème en considérant un travail qui risque de ne jamais être traité ; par rapport à ce travail critique, on partitionne les travaux en deux catégories : ceux qui doivent être traités auparavant en vertu de l'ordonnancement décidé par l'algorithme de "Safety", et les autres. Ces autres travaux ne recevront aucune nouvelle allocation susceptible de délayer davantage le travail critique.
- b) Elle ne règle pas le problème de choix des travaux à démarrer et, plus généralement, le problème d'efficacité : si plusieurs travaux sont susceptibles de recevoir leurs ressources, lequel choisir d'abord ?

TROISIEME PARTIE

STRATEGIES D'ALLOCATION DE RESSOURCES DANS UN RESEAU D'ORDINATEURS

GENERALITES

Dans cette partie, nous essayons de déterminer des structures d'algorithmes d'allocation de ressources non-préemptibles dans un réseau hétérogène d'ordinateurs qui permettent d'éviter des situations d'étreintes fatales, tout en préservant à chaque noeud du réseau son indépendance de décision. Le problème d'efficacité ne sera examiné que de façon annexe et sans aucune prétention.

PROBLEMES D'ALLOCATION DANS UN RESEAU.

Comment un réseau d'ordinateurs se distingue-t-il d'un simple ordinateur partagé en ce qui concerne les algorithmes de micro et de macro allocations ?

1°) Décentralisation de la fonction d'allocation. Alors que dans un ordinateur partagé les décisions d'allocation sont prises par un allocateur central qui connaît tout l'état du système (ressources et processus), il ne peut en être ainsi dans un réseau d'ordinateurs pour plusieurs raisons :

- a) Chaque centre de calcul optimise son propre rendement (et non le rendement global) ;
- b) Les délais de transmission augmenteraient considérablement l' "overhead" (temps nécessaire pour prendre la décision) ;
- c) Le travail en temps réel d'un allocateur central ne serait de toute façon pas compatible avec l'existence d'un énorme réseau dont le nombre de processus ou de ressources peut tendre vers l'infini (et évolue constamment).

2°) Une nouvelle catégorie de ressources apparaît : c'est le réseau des transmissions de messages entre ordinateurs.

I - SECURITE VIS A VIS DES ETREINTES FATALES DANS UN RESEAU D'ORDINATEURS

Que peut signifier la notion d'étreinte fatale dans un réseau d'ordinateurs ?

- 1°) Etreinte fatale au niveau du sous-système de transmission (1) : les messages ne peuvent plus progresser dans le réseau faute de mémoire disponible aux noeuds (2). Ce problème ne sera pas abordé dans ce papier ; le système de transmission sera supposé être opérationnel.
- 2°) Etreinte fatale dans le réseau considéré comme un ensemble d'ordinateurs : aucun processus ne peut plus progresser dans le réseau ; tous les ordinateurs du réseau sont bloqués.
- 3°) Un ordinateur particulier du réseau est bloqué momentanément en attendant que le traitement de certains processus sur certains autres ordinateurs lui permette de reprendre son travail.

Nous étudierons comment éviter les étreintes fatales en nous plaçant dans l'une des deux hypothèses suivantes :

[H1] Le réseau est faiblement "safe" si il existe un ordre dans lequel on peut traiter l'ensemble des travaux du réseau, libérant les ressources utilisées par les travaux au fur et à mesure de leur terminaison.

Cette hypothèse implique qu'au moins un ordinateur du réseau travaille à un instant donné.

[H2] Le réseau est fortement "safe" si chacun des ordinateurs du réseau est "safe" quelque soit l'état (à priori ignoré par lui) des autres ordinateurs du réseau. Un réseau peut aussi être localement fortement safe, c'est-à-dire qu'un ordinateur déterminé est safe, quel que soit le comportement des autres.

Dans un réseau fortement "safe", tous les ordinateurs du réseau fonctionnent simultanément à un instant donné.

(1) Nous supposons ici une transmission par paquets semblable à celle du réseau ARPA.

(2) C'est-à-dire aux IMP's (Interface Message Processors)

A - SYSTEME DISTRIBUE D'ORDONNANCEMENT ARBITRAIRE DES TRAVAUX "RESEAUX" CONNU PAR LES NOEUDS.

Dans un réseau faiblement "safe", il faut (et il suffit) qu'il existe un ordre de traitement des travaux. Comment déterminer un tel ordre ?

Cet ordre peut être déterminé comme une cascade d'ordres partiels (cf. fig 9).

Distinguons, pour un ordinateur donné, trois types de travaux : les travaux "locaux" qui n'utilisent que l'ordinateur local, les travaux "communs" qui utilisent à la fois l'ordinateur local et d'autres ordinateurs du réseau, et enfin les travaux "lointains" qui n'utilisent que d'autres ordinateurs. L'ordinateur local doit être assuré que l'ordre dans lequel il pourra traiter les travaux "communs" sera le même ordre dans lequel lesdits travaux pourraient être traités sur d'autres ordinateurs du réseau. Par contre, l'ordre des travaux "lointains" lui est indifférent, et l'ordre des travaux "locaux" ne concerne que lui.

Nous pouvons alors imaginer la situation suivante : un ordonnateur "réseau" détermine un certain ordre de traitement possible pour les travaux qui utilisent plusieurs ordinateurs. Chaque ordinateur sera informé de cet ordre partiel et devra le respecter lorsqu'il cherchera un ordre total de ses processus locaux ou communs qui le rende "safe". Chaque ordinateur sera donc responsable d'étendre l'ordre partiel en y incorporant ses processus locaux. Il pourra réaliser cette extension de deux façons :

- 1°) En cherchant une situation faiblement "safe" ; dans ce cas, n'importe quelle combinaison des passages des travaux respectant l'ordre partiel réseau pourra convenir.
- 2°) En cherchant une situation fortement "safe" ; dans ce cas, il cherchera un ordre total qui fasse passer les travaux "réseaux" après tous les travaux locaux. Ceci implique simplement de trouver un ordre "safe" des travaux locaux, et un autre ordre "safe" des travaux réseaux.

Cette méthode devra cependant être tempérée par la possibilité d'imposer une priorité à certains travaux réseau, afin d'éviter qu'ils ne soient délayés indéfiniment. Il semble d'ailleurs nécessaire de permettre à un travail réseau de "réserver" certaines ressources sur différents ordinateurs, à condition de demander cette réservation suffisamment à l'avance et d'être disposé à payer un prix suffisant (voir 4ème partie).

Dans certains cas, il est possible de considérer le réseau comme étant constitué de sous-réseaux suffisamment indépendants. La figure 11 montre un exemple où l'ordonnancement des travaux passant sur W et X a lieu indépendamment de l'ordonnancement des travaux pouvant utiliser X, Y et Z. Les travaux susceptibles d'utiliser des ordinateurs appartenant aux deux sous-réseaux doivent s'adresser à un ordonnanceur global, qui prend donc les décisions d'ordonnancement ne pouvant pas relever des niveaux au-dessous de lui.

Remarquer qu'une situation telle que celle de la figure 12 est interdite. En effet, des travaux utilisant à la fois X et Y pourraient recevoir des ordres différents par les ordonnanceurs respectifs de X et Y. Par contre, le schéma de la figure 13 est permis.

Modification de la structure des ordonnanceurs.

Les places des ordonnanceurs sont déterminées en vue de n'imposer qu'un ordre partiel aussi peu contraignant que possible, de façon à laisser un maximum de liberté aux ordinateurs noeuds du réseau dans l'ordonnancement des travaux. Cependant, des modifications des conditions de la demande peuvent nécessiter une restructuration des ordonnanceurs. Cette restructuration peut intervenir instantanément, dans la mesure où elle n'affecte pas l'ordre des travaux déjà ordonnancés et où les nouveaux travaux passeront après ceux qui avaient été ordonnés par l'ancienne structure.

Il semblerait d'ailleurs que des conditions plus souples puissent être adoptées à l'occasion de la réorganisation de la structure : c'est l'un des nombreux points qui devront faire l'objet de travaux ultérieurs.

B - PROCEDE D'ORDONNANCEMENT PARTIEL DES TRAVAUX DU RESEAU PERMETTANT AUX ORDINATEURS NOEUDS DU RESEAU D'IGNORER CET ORDONNANCEMENT.

La méthode du paragraphe précédent imposait aux ordinateurs du réseau de respecter certains ordres partiels déterminés par les ordonnanceurs "réseau". Il peut cependant être intéressant de ne pas imposer aux ordinateurs particuliers de connaître cet ordre, et de considérer le "réseau" dans son ensemble comme un utilisateur particulier des différents ordinateurs du réseau. Cette situation est schématisée dans la figure 14.

Un pseudo-allocateur est à la fois un pool d'utilisateurs, un pool de ressources maximum pouvant être obtenues en un temps fini, et un pool de ressources couramment allouées à ses utilisateurs.

Un ordinateur réel est un pseudo-allocateur dont les ressources maximum sont identiques aux ressources physiques réellement disponibles.

Un pseudo-allocateur α peut être le client d'un ou plusieurs autres pseudo-allocateurs β . Dans ce cas, la quantité de ressources allouées à α sur β est identique à la somme des ressources allouées par α à ses propres usagers. Mais la somme des ressources allouables par α à l'ensemble de ses usagers en un temps fini peut être nettement supérieure à la quantité maximum des ressources que α pourra obtenir sur β . En d'autres termes, soit :

$\vec{A}$ le vecteur des ressources allouées à α

$\vec{B}$ le vecteur des ressources allouables à α

$\vec{A}_i, \vec{B}_i$ les vecteurs alloués et allouables à l'utilisateur n° i de α

alors on a :

$$\vec{A} = \sum \vec{A}_i$$
$$\forall i, \vec{B} \geq \vec{B}_i$$

(mais, éventuellement, $\vec{B} < \sum \vec{B}_i$) et de plus, le pseudo-allocateur α doit s'assurer que la situation formée par les $\vec{A}_i$ et les $\vec{B}_i$ est "safe" compte tenu de ses propres $\vec{A}$ et $\vec{B}$.

Dans la hiérarchie arborescente des pseudo-allocateurs, chacun doit s'assurer que ses promesses à ses clients le rendent "safe" compte tenu des ressources maximum auxquelles il a droit. Remarquez qu'un schéma analogue à celui de la figure 12 est toujours interdit, pour les mêmes raisons qu'au paragraphe A.

Lorsqu'un travail aura besoin d'un ensemble d'ordinateurs, il s'adressera au pseudo-allocateur, le plus bas dans la hiérarchie qui ait au-dessous de lui cet ensemble d'ordinateurs sur lequel des ressources non-préemptibles sont désirées.

Le schéma du § B a certains avantages sur le schéma du § A. D'abord, la séparation des tâches entre ordonnanceurs est plus nette. Ensuite, il ne nécessite aucun algorithme spécial permettant de chercher un ordre total à partir de certains ordres partiels. Il permet aussi éventuellement, aux pseudo-allocateurs de réserver de façon permanente certaines ressources sur leurs fournisseurs de niveau inférieur, ce qui permet de garantir une certaine efficacité aux travaux du réseau.

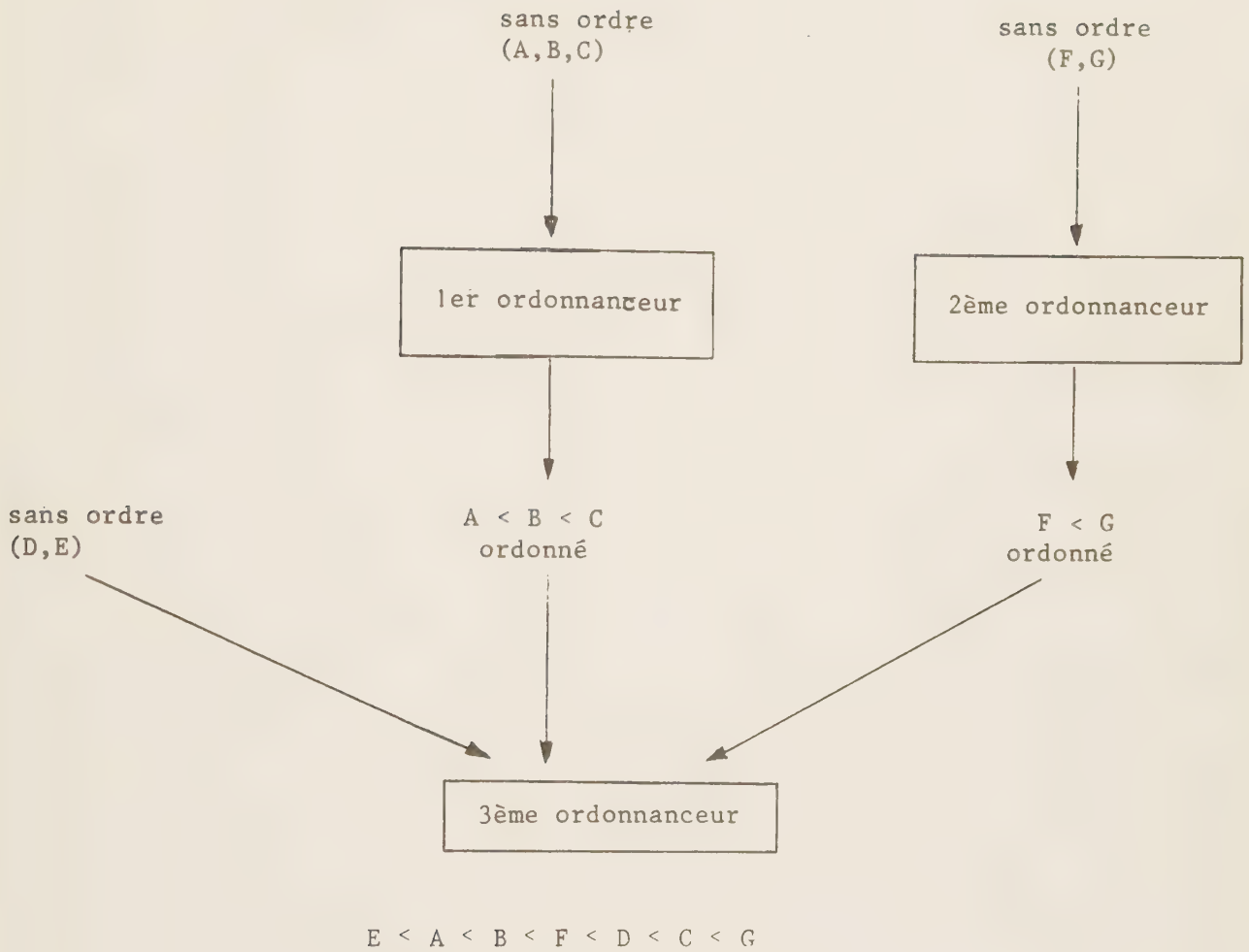


Figure 9

Ordonnancement par étapes

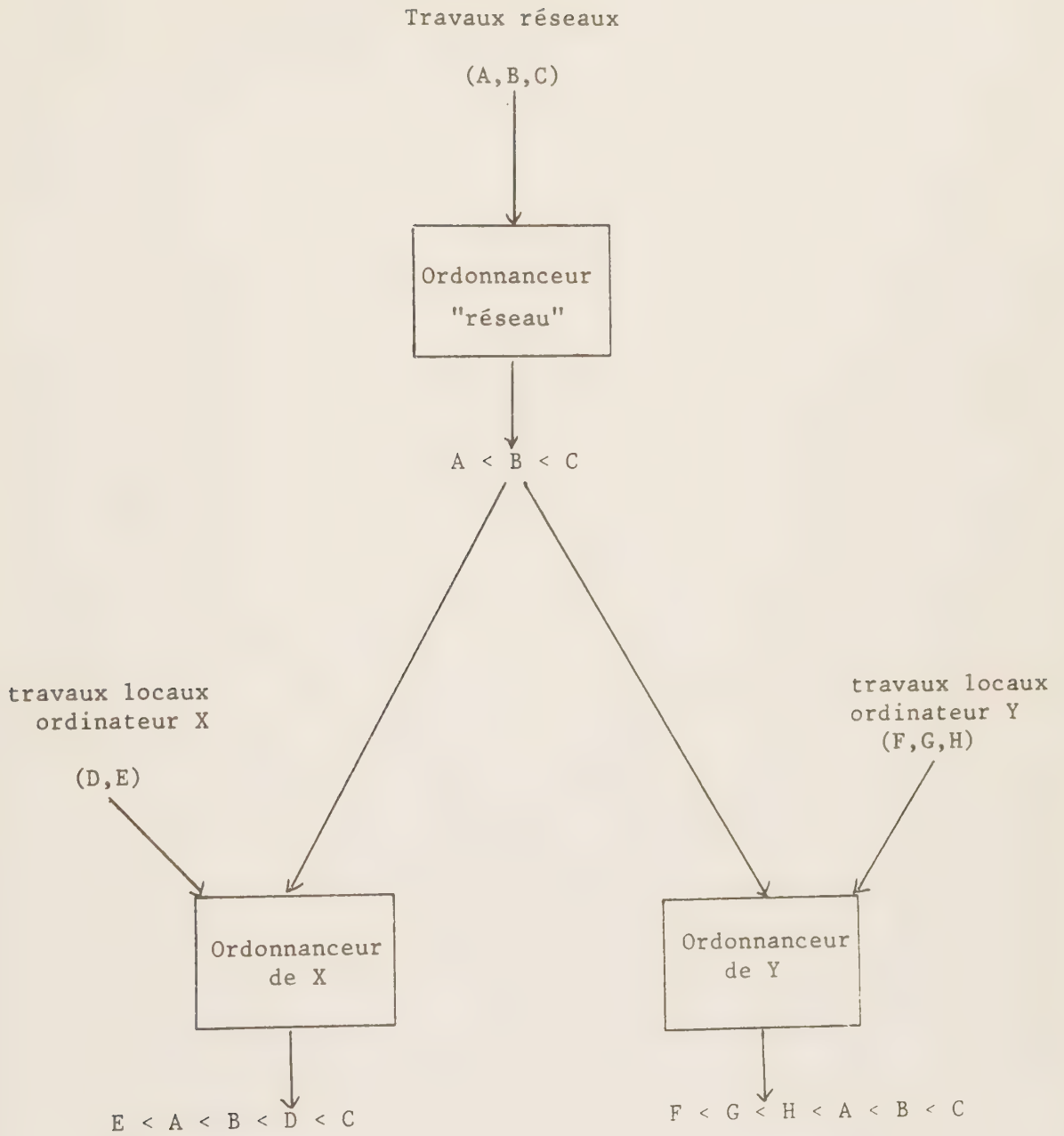


Figure 10

Ordonnanceur réseau centralisé

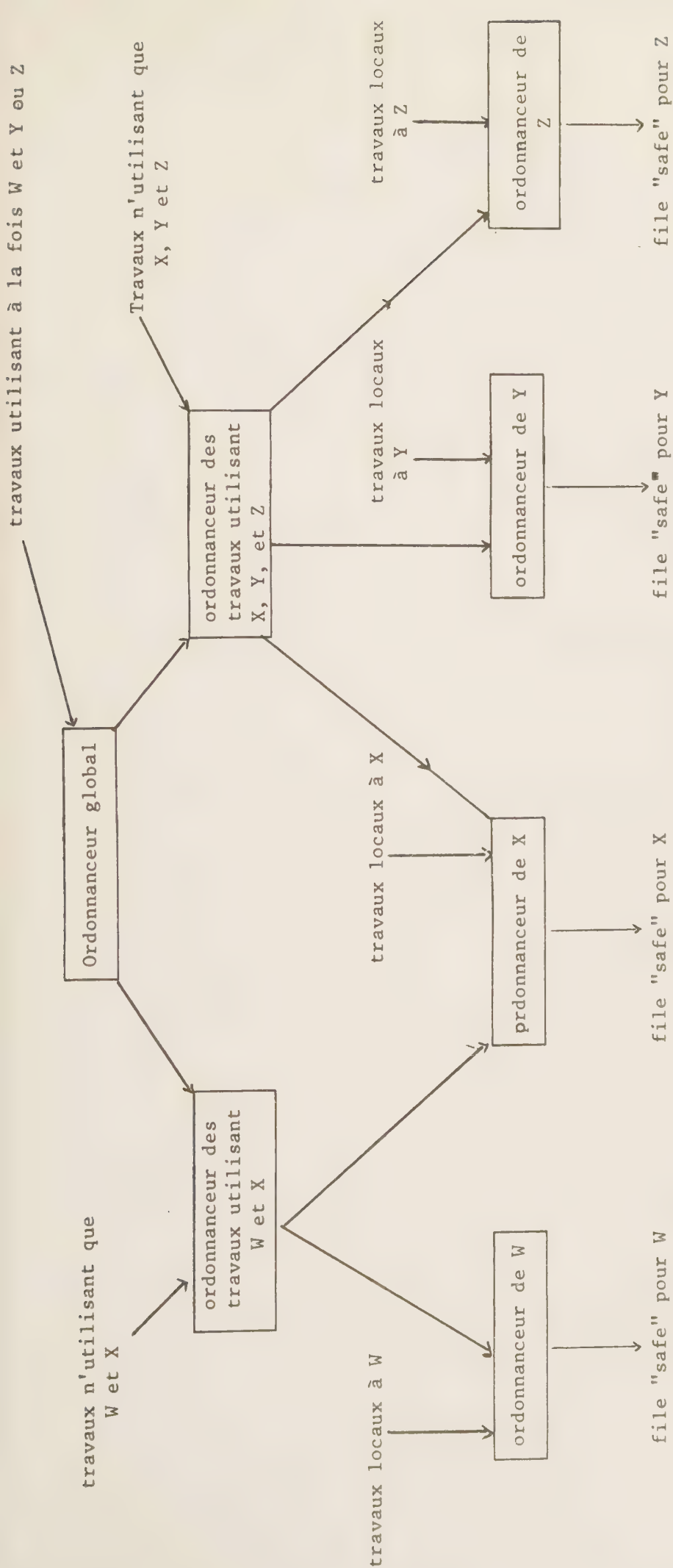


Figure 11 - Hiérarchie d'ordonnanceurs

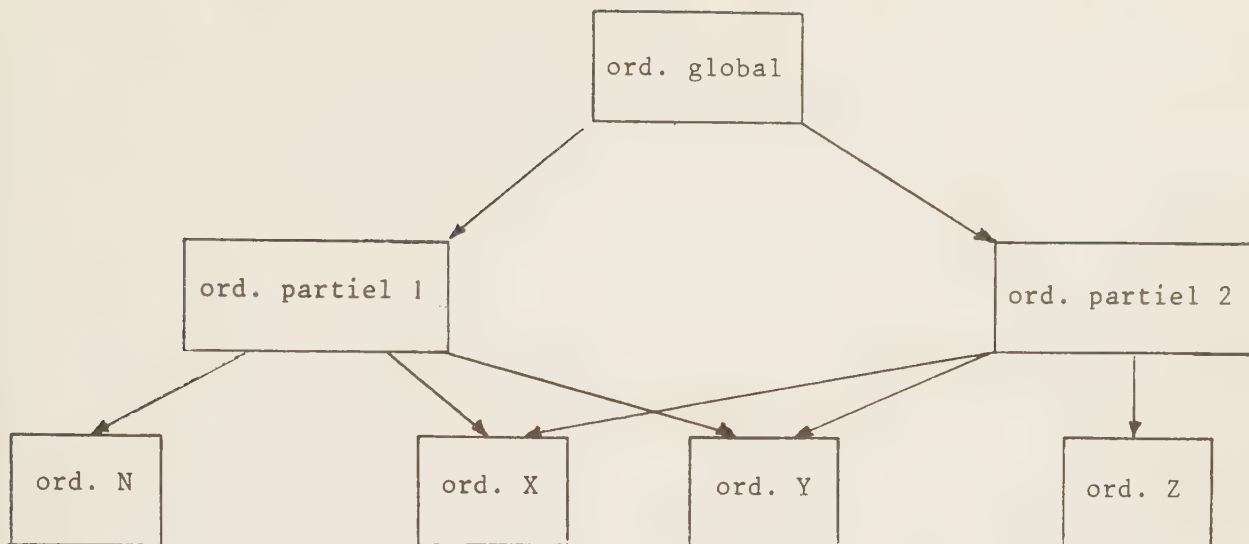


Figure 12 - Configuration interdite

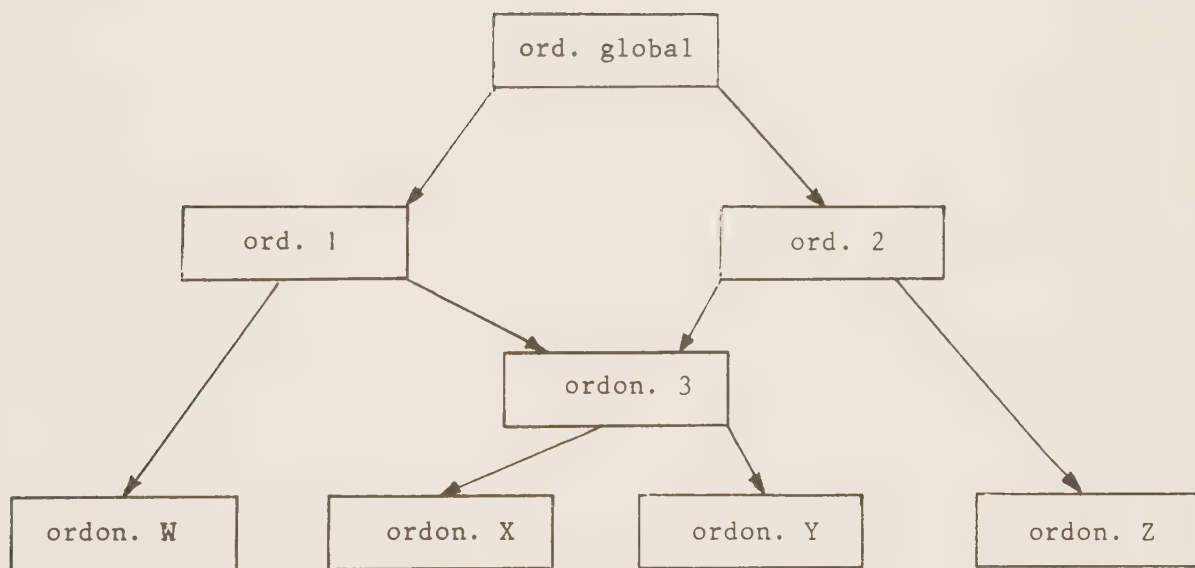


Figure 13 - Configuration autorisée.

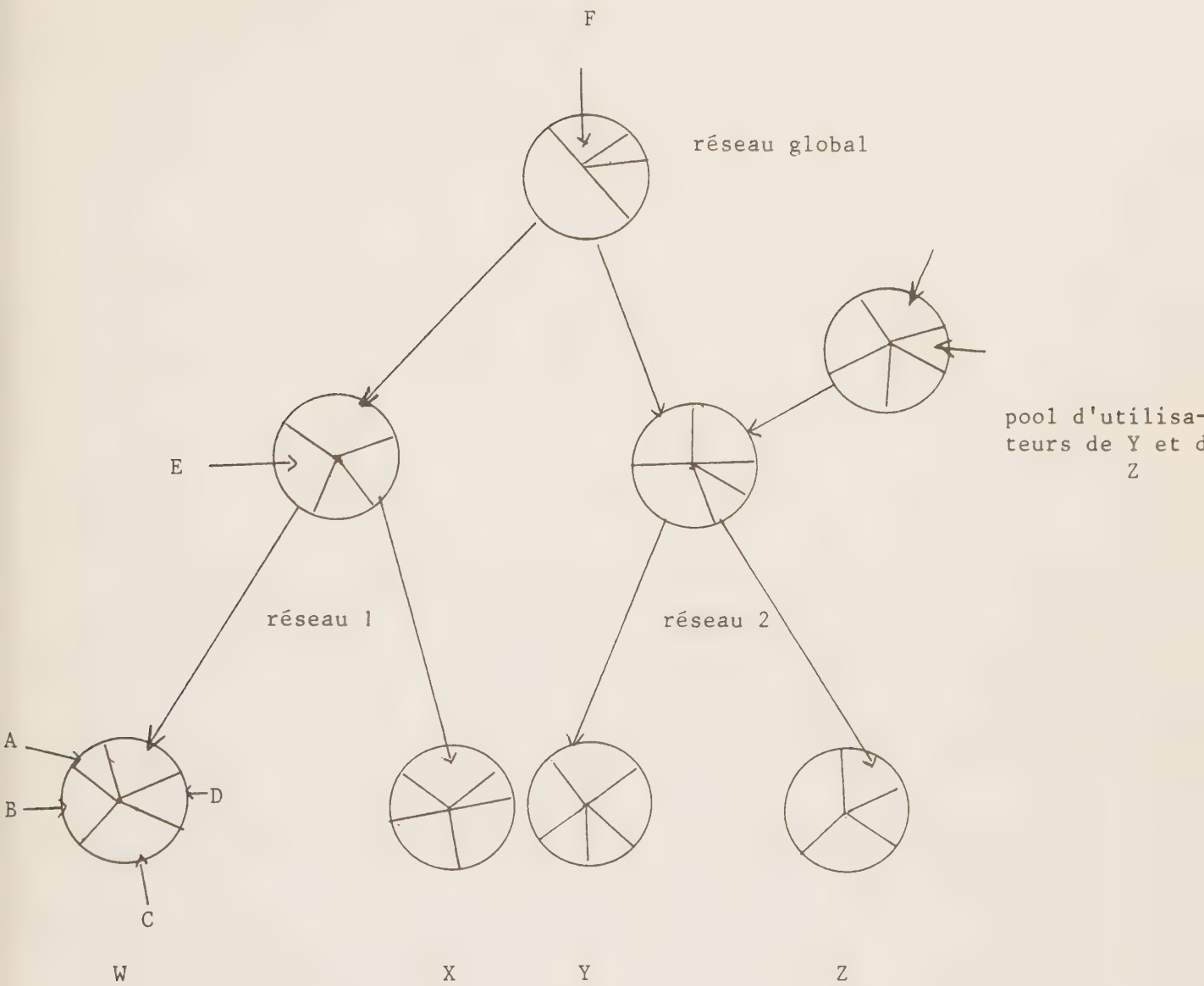


Fig. 14 - Hiérarchie des pseudo-allocateurs

Les différents ordinateurs du réseau ont un client particulier qui est un pool d'utilisateurs, qui peut lui-même avoir comme client un autre pool d'utilisateurs.

QUATRIEME PARTIE

PROBLEME DE RESERVATIONS DES RESSOURCES PAR LES UTILISATEURS ;
ETUDE PAR SIMULATION

- GENERALITES

Dans cette deuxième partie, on envisage le cas où un travail réseau a la possibilité de "réserver" certaines ressources sur différents ordinateurs. Cette possibilité semble être nécessaire si l'utilisateur souhaite d'être assuré d'avance du déroulement de son travail et s'organiser en conséquence. Ainsi, pour exécuter une tâche donnée, l'utilisateur dépose une demande de réservation au réseau. Dans sa demande, il indique :

- les ressources désirées sur différents ordinateurs (avec la quantité demandée sur chacune),
- le temps auquel il souhaite commencer à travailler : TDEB
- la durée de son travail : DURE
- le temps limite de démarrage de son travail (temps après lequel le travail n'a plus aucun intérêt) : TLIM
- un coefficient qui peut être considéré comme le prix qu'il est prêt à payer pour sa demande : COEF

On voit qu'on a ici un problème de réservation ou "rendez-vous". C'est un problème très général qui se pose chaque fois que différents utilisateurs veulent se réserver pendant un certain intervalle de temps, l'emploi d'un certain nombre de ressources communes. Dans le cas des réseaux d'ordinateurs il faut que

le réseau soit capable, compte tenu de la capacité de chaque ressource et des intervalles de temps où elle est demandée, d'organiser des "rendez-vous" entre les processus et les ressources. Pour cela, il faut établir des "carnets de rendez-vous" dans lesquels sont inscrites les demandes de réservations qui sont acceptées (selon un critère donné) parmi l'ensemble des demandes déposées.

I - CRITERES D'ALLOCATION DES RESSOURCES

Le programme allocateur doit généralement satisfaire deux types de critères au cours de l'allocation de ses ressources :

a) Un critère impératif :

Il faut que pour chaque ressource et dans chaque intervalle de temps, la somme des quantités exigées par les demandes acceptées soit compatible avec la capacité totale de la ressource correspondante. En d'autres termes, il ne faut jamais promettre de donner plus que la quantité disponible.

b) Un critère d'efficacité :

On cherchera en allouant les ressources à maximiser une certaine fonction de satisfaction dont on définira par la suite.

Ainsi, notre programme de simulation cherche dans une première version à satisfaire le premier critère. Dans une deuxième version, tout en satisfaisant le critère impératif, il cherche à maximiser la fonction de satisfaction, donc à optimiser la procédure d'allocation des ressources.

I - STRUCTURE DU PROGRAMME DE SIMULATION

Le programme est composé en gros de deux parties : l'analyseur des données de simulation et l'allocateur des ressources proprement dit. En effet, pour simuler l'allocation des ressources dans un réseau d'ordinateur, il faut se donner d'une part un réseau, d'autre part un certain nombre de demandes de réservations. Les données du programme ont pour rôle d'une part de décrire d'une façon précise le réseau en énumérant les ordinateurs participants avec leurs ressources respectives et la capacité de chaque ressource, d'autre part, de préciser les demandes de réservation de ces ressources et leurs paramètres (voir 3.1).

3.1. LANGAGE DE DONNEES

Le langage utilisé pour décrire le réseau et les demandes est simple et précis. Le vocabulaire de base est formé de lettres, de chiffres, de quelques symboles et quelques mots-clé.

- les symboles utilisés sont $\{ / : (,) * = \& \}$
- les mots-clé sont $\{ \underline{\text{RESS}}, \underline{\text{REQU}}, \underline{\text{TDEM}}, \underline{\text{TDEB}}, \underline{\text{DURE}}, \underline{\text{TLIM}}, \underline{\text{COEF}} \}$

Ces mots-clé doivent être utilisés obligatoirement car ils permettent au programme d'identifier certains paramètres de la simulation. L'utilisateur peut construire avec le vocabulaire ci-dessus et d'après certaines règles des entiers et des identificateurs. Ces règles de grammaire s'expriment sous forme de Backus de la façon suivante :

$\langle \text{entier} \rangle :: = \langle \text{chiffre} \rangle | \langle \text{entier} \rangle \langle \text{chiffre} \rangle$

$\langle \text{chiffre} \rangle :: = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

$\langle \text{identificateur} \rangle :: = \langle \text{lettre} \rangle | \langle \text{identificateur} \rangle \langle \text{lettre} \rangle | \langle \text{identif.} \rangle \langle \text{chiffre} \rangle$

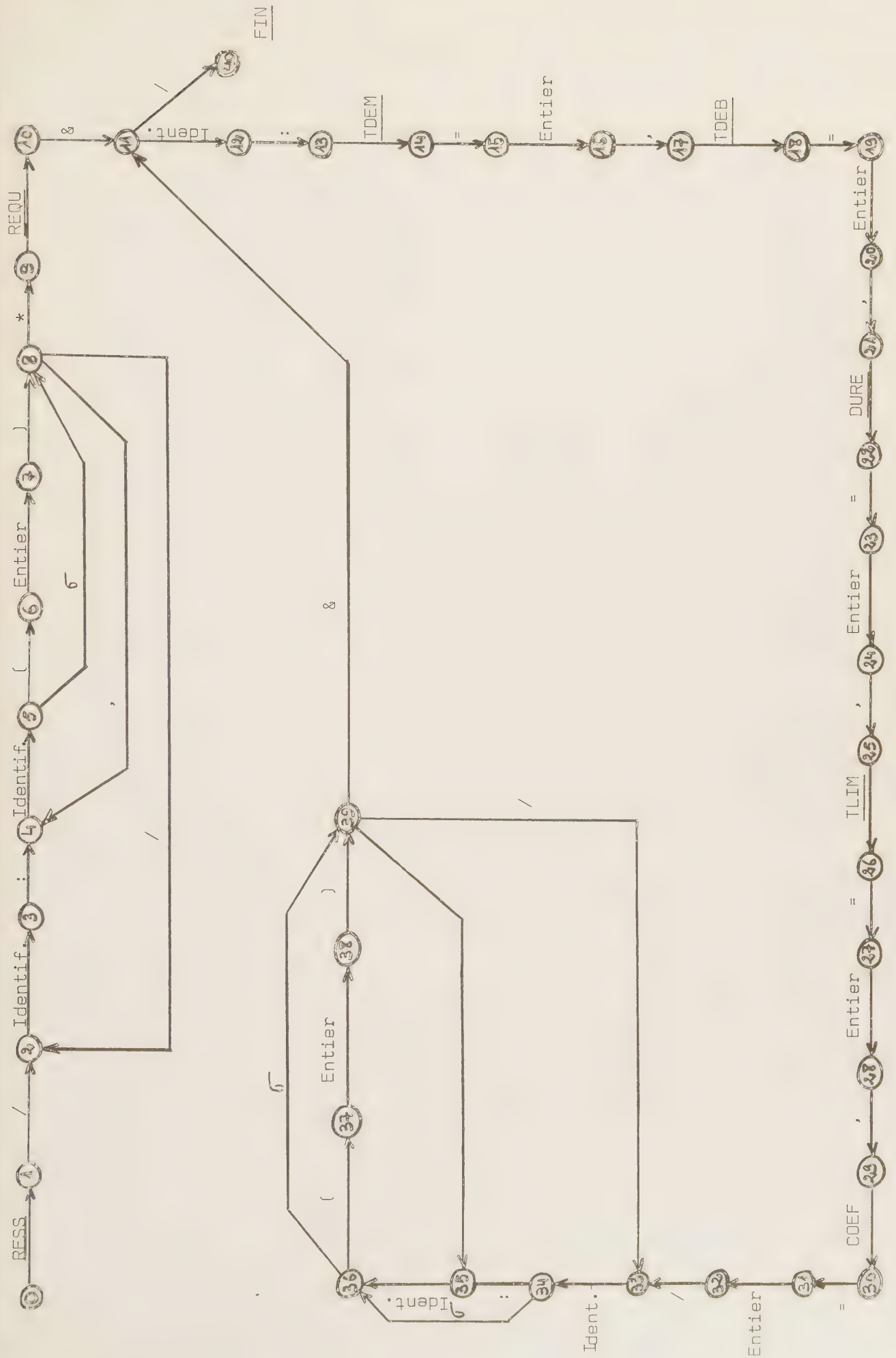
$\langle \text{lettre} \rangle :: = A | B | C | \dots | Z$

CONTRAINTES :

La longueur maximum des identificateurs est fixée à 8 caractères et celle des entiers à 5 chiffres.

3.2. ANALYSEUR SYNTAXIQUE

Le langage est défini complètement par sa grammaire qui est représentée graphiquement par un automate d'état fini : (voir le schéma de l'automate). L'analyseur syntaxique vérifie la structure des données et ne garde que les demandes qui ont été formulées correctement.



3.3. STRUCTURE DE PROGRAMME D'ALLOCATION

Nous avons vu au paragraphe II qu'il existe deux versions différentes du programme d'allocation des ressources. On peut donc allouer les ressources suivant deux politiques distinctes, dont la première satisfait le critère a et la seconde les critères a et b. Les deux versions ont donc une grande partie commune et comme on verra par la suite la différence qui les sépare est en fait dans l'ordre dans lequel elles examinent les demandes. Pour satisfaire le critère impératif a, la partie commune et essentielle des deux versions doit connaître dans le temps l'état d'occupation de chaque ressource du réseau. Pour cela, on utilise pour chaque ressource une liste chaînée. Cette liste est une suite d'intervalles de temps à chacun desquels est affecté un taux d'occupation global. Cette structure de données convient particulièrement car il s'agit surtout d'opérations d'insertions d'intervalles ou de mise à jour (modification de la longueur de la chaîne). Etant donné une demande, (les numéros d'ordre des demandes sont en général différents dans les deux versions), il faut vérifier dans les listes si l'insertion est possible. L'organigramme général de la recherche dans les listes est donné ci-dessous.

En résumé, on peut dire que l'organigramme cherche à satisfaire une demande le plus tôt possible, à partir de TDEB (temps auquel l'utilisateur souhaite commencer à travailler) et dans l'intervalle du temps (TDEB, TLIM) où TLIM est le temps limite de démarrage du travail.

A) Première version : METHODE NON-OPTIMISEE

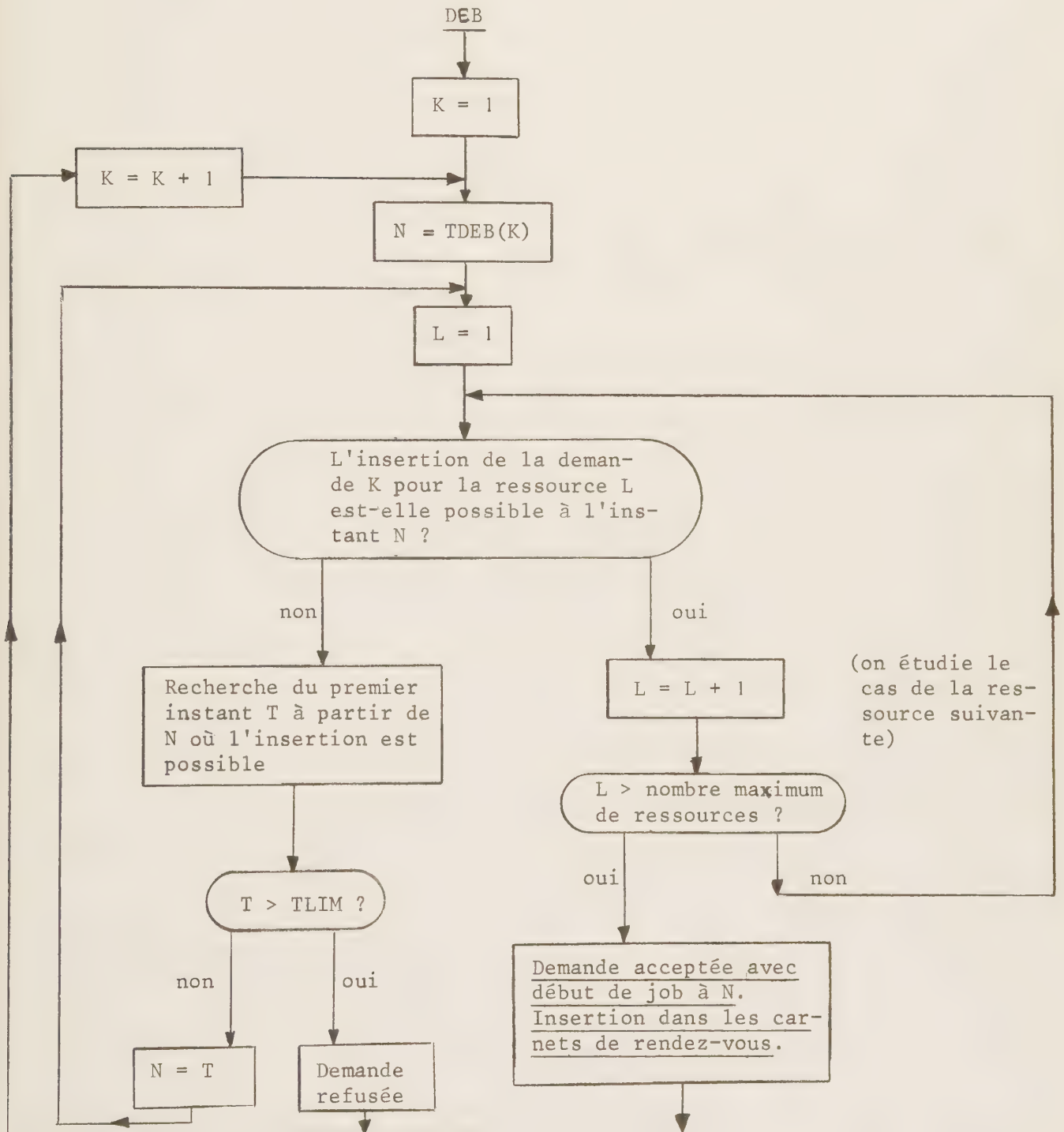
Dans cette première méthode, on examine les demandes dans l'ordre d'arrivée. A chaque demande est affecté un coefficient TDEM qui est le temps simulé où la demande a été faite. L'algorithme de recherche décrit précédemment est donc appliqué aux demandes dans l'ordre des TDEM croissants.

B) Deuxième version : METHODE OPTIMISEE

Dans la deuxième méthode, on a essayé d'optimiser le fonctionnement de système, en examinant les demandes non pas dans l'ordre d'arrivée, mais suivant un certain ordre de priorité. Cela signifie qu'en temps réel,

K : numéro d'ordre de prise en compte de la demande

L : numéro de ressource demandée par K



Remarque : Les itérations portent sur L et N. La recherche s'arrête dans deux cas :

- $L > \text{nombre maxi de ressource}$ → demande acceptée
- $N > TLIM$ → demande refusée.

l'utilisateur n'obtient pas tout de suite une réponse à sa demande mais après un certain temps d'attente. En effet, pour optimiser son fonctionnement, le réseau peut prendre une décision à des intervalles de temps prédéfinis et sur des "paquets de demandes" déposés pendant le dernier intervalle. Dans notre simulation, l'intervalle de temps choisi est l'intervalle total de simulation.

Quant à la notion de priorité d'un utilisateur, il est bien évident qu'elle peut être définie de nombreuses façons. On la définit ici en faisant appel à d'autres notions telles que : "le coût d'une ressource", "le poids d'un utilisateur", ou la priorité d'un utilisateur". Ces notions restent à définir :

Le coût d'une ressource est un coefficient qui est calculé en fonction de l'occupation de cette ressource. Si :

C_i = coût de la ressource

Q_{ij} = Quantité de ressource i demandée par l'utilisateur j

T_j = durée de travail de l'utilisateur j

On a :

$$C_i = \sum_j Q_{ij} * T_j$$

En d'autres termes plus une ressource est demandée plus son coût est élevé.

Le poids d'un utilisateur est proportionnel à la quantité totale de ressource qu'il a demandé.

Poids d'un utilisateur j :

$$P_j = \sum_i Q_{ij} * C_i$$

Enfin, la priorité D_j d'un utilisateur est calculée de la façon suivante :

$$D_j = \text{COEF}(j) / (P_j * T_j)$$

$\text{COEF}(j)$ est un coefficient affecté à la demande de l'utilisateur j et peut être considéré comme le prix qu'il est prêt à payer pour sa demande. Ainsi, la priorité de l'utilisateur est d'autant plus élevée que D_j est plus grand. L'algorithme de recherche ci-dessus sera appliqué alors sur les demandes rangées suivant les D_j décroissant.

7 - RESULTATS

On a fait une simulation de réservation des ressources avec plusieurs ensembles de demandes (voir les exemples donnés). Sur chaque ensemble sont appliquées les deux politiques définies précédemment. Dans chaque cas, sont donnés : l'image des cartes de données de simulation ; les listes et les nombres des demandes acceptées et refusées ; un tableau qui montre l'état d'occupation de chaque ressource du réseau dans le temps, enfin la valeur finale de la fonction de satisfaction qui peut être interprétée comme le gain du réseau et qui n'est autre chose que la somme des coefficients (COEF) des demandes acceptées.

- APPLICATIONS

A part les réseaux d'ordinateurs, le programme de réservations trouve un champ d'application chaque fois qu'il est question de réserver pendant un certain temps quelques ressources. On peut imaginer par exemple l'application que peut avoir ce programme dans une compagnie de location de voitures. Les agences de cette compagnie (ou réseau) sont équivalentes aux ordinateurs du réseau. Chaque agence dispose d'un certain nombre de catégories ou marques de voitures (équivalent aux différents types de ressources). Le nombre de voitures dans chaque catégorie est équivalent à la quantité de chaque ressource. Ainsi un client ou une société peut demander la réservation pendant une certaine durée de plusieurs voitures de différentes catégories disponibles aux différents endroits. En décrivant le réseau des agences et les demandes, le problème peut être résolu par le programme de réservations.

CONCLUSION GENERALE

On constate sur les exemples donnés que dans tous les cas la deuxième méthode (méthode optimisée) permet d'atteindre une valeur de la fonction de satisfaction plus élevée que dans la première méthode. Le rapport des valeurs trouvées par la fonction de satisfaction dans les deux cas dépend évidemment des valeurs numériques attribuées aux paramètres des demandes. On ne peut donc pas comparer d'une façon précise les deux méthodes, mais seulement affirmer la supériorité de la deuxième méthode par rapport à la première. En effet, pour trouver la valeur optimale de la fonction de satisfaction, il faut essayer tous les ordres possibles d'examen des demandes. Ce qui exige pour un nombre de demandes élevé un temps trop long. On voit ainsi que le choix des méthodes approximatives est justifié.

En ce qui concerne les stratégies d'allocation de ressources dans un réseau, il faut noter que les schémas développés dans la troisième partie semblent plus attrayants que des schémas antérieurement utilisés comme celui de Havender [5] où les processus examinaient les ordinateurs et les ressources toujours dans le même ordre afin d'éviter les embrasses mortelles. Mais beaucoup de progrès restent encore à réaliser dans notre compréhension des problèmes d'allocation.

D'abord, quels critères permettent d'estimer des algorithmes d'allocation dans un réseau ? Comment se comparent les différentes méthodes d'allocation vis à vis de ces critères ? Comment caractériser les graphes autorisés ou non autorisés de rapports des pseudo-allocateurs entre eux ?

Quel schéma adopter pour la réservation de ressources non-préemptibles ?

Comment traiter les ressources préemptibles en vue de leur utilisation à distance ?

Quel mécanisme de prix de ressources y-a-t-il lieu d'institutionnaliser sur le réseau ?

On espère que des recherches semblables pourront apporter des éléments aux praticiens qui écrivent des systèmes d'exploitation devant éventuellement s'intégrer dans un réseau d'ordinateur.

RESS / IR360:MC1(128),LEC1,DISQ1(5),IMP1(2)/IR50 :MC2 (256),LECC2,BND2(7),
 DISQ2(6)/P1400 : MC3(256),DISQ3(4),ECR3(3),IRAC3(2),PROG3(50) / PDP : MC4(64),
 IMP4,LECC4 / IR60 : MRAP(1000),BNQ(4) / ICL : DISQ6(6) ,MMAS(512)
 *REGU &GASTON:TDEN=5,TDEN=20,DURE=91,TLIM=30,COEF=6/IR60:MRAP(50),BNQ(2)
 &ROBERT:TDEN=6,TDEN=10,DURE=400,TLIM=300,COEF=123/IR360:MC1(64),DISQ1/IR
 MC2(64),BND2(4) 50:
 &ALPHA:TDEN=7,TDEN=12,DURE=234,TLIM=200,COEF=95/ICL:DISQ6(3),MMAS(200) /IR360:
 MC1(64),LEC1,DISQ1,IMP1 /P1400:MC3(64)
 &GAMMA:TDEN=7,TDEN=12,DURE=234,TLIM=200,COEF=95/PDP:MC4(12),IMP4
 /IR360:MC1(40),DISQ1(1),LEC1/P1400:ECR3(2),DISQ3(1),MC3(10)&VICTOR:TDEN=8
 ,TDEN=15,DURE=75,TLIM=70,COEF=2/ICL:DISQ6(2)/IR50:MC2(30),BND2(2),
 DISQ2(2),LECC2 / IR60 :MRAP(100),BNQ(1) /IR360 : MC1(12),DISQ1(2)
 &ROGER: TDEN=20,TDEN=30,DURE=40,TLIM=70 ,COEF=12 /IR60:MRAP(50)/PDP: MC4(8)
 / IR360: MC1 (4) ,DISQ1(1)/IR50 :MC2(15) ,BND2(2) /P1400 :MC3 (20) /ICL:MMAS
 (46)&OSCAR: TDEN=40 ,TDEN=115 ,DURE=50,TLIM=200, COEF=20 /IR360:MC1(64),
 DISQ1(2),IMP1(1)/P1400:IRAC3 &PIERRE: TDEN=41,TDEN=60,DURE=86,TLIM=65,COEF=9
 /IR60 : MRAP(20),BNQ / ICL:MMAS (150) / P1400: MC3(12) ,PROG3(10) & JUJU :
 TDEN=80 ,TDEN=90,DURE=53,TLIM=135,COEF=30 / IR360 : MC1(15) /IR50 :MC2 (14)
 BND2,DISQ2 / P1400:MC3(20) ,PROG3(2) /PDP:MC4(2) & DUPONT :TDEN=81 ,TDEN=85
 DURE=40,TLIM=300,COEF=22/ICL:MMAS(101),DISQ6/IR60:MRAP(5)/PDP:MC4(12)
 /P1400 :MC3(14),IRAC3 & TOU : TDEN=120, TDEN=150,DURE=30 ,TLIM=200,COEF=5/
 ICL:MMAS(26) / IR60 : MRAP(5) /PDP:LECC4 /IR360 :MC1 (16) /IR50 :BND2
 & MARTIN : TDEN=122 ,TDEN=170,DURE=91,TLIM=301,COEF=33/ IR360 :MC1(45),DISQ1
 /IR50 :MC2(10) /P1400 :MC3(33),PROG3(3) /ICL :MMAS(17),DISQ6 /IR60:MRAP(70)
 & PAUL : TDEN=122 ,TDEN=171,DURE=184,TLIM=190 ,COEF=195/ IR360:MC1(4)/
 IR50:MC2(11),BND2 /P1400 : MC3(17) ,PROG3(11)/ICL :MMAS(11)&LOULOU :
 TDEN= 123 ,TDEN=250,DURE=100,TLIM=400,COEF=34 /P1400: PROG3(3)/PDP : MC4(2)
 /IR360:MC1(4),DISQ1(1)/IR50:MC2(15),BND2(2)/P1400:MC3(20)/ICL:MMAS/IR60:MRAP(2)
 &BOR :TDEN=130,TDEN=150,DURE=100,TLIM=300,COEF=50/ IR360: MC1(80),DISQ1(2),IMP1
 /IR50:MC2(128)/MC3(5),ECR3
 &HORIS:TDEN=132,TDEN=150,DURE=66,TLIM=305,COEF=55/PDP:MC4(30),IMP4/MRAP(200),BNQ
 / IR360: MC1 (4) ,DISQ1(1)/IR50 :MC2(15) ,BND2(2) /P1400 :MC3 (20) /ICL:MMAS
 (1) / ICL : DISQ6(2),MMAS(100)
 &DURANT : TDEN=133,TDEN=142,DURE=74,TLIM=325,COEF=45/IR360:MC1(20),DISQ1(2),LEC1
 /MC2(10),MMAS (500),BNQ(1) /PDP :MC4(32)
 &JULES:TDEN=136,TDEN=150,DURE=70,TLIM=240,COEF=20 /IR60:MRAP(50)/PDP: MC4(8)
 / IR360: MC1 (4) ,DISQ1(1)/IR50 :MC2(15) ,BND2(2) /P1400 :MC3 (20) /ICL:MMAS &
 JOEL:TDEN=137,TDEN=161,DURE=70,TLIM=350,COEF=45/IR360:MC1(64),LEC1/IR50:MC2(256)
 &DUPANLOU:TDEN=137,TDEN=145,DURE=69,TLIM=325,COEF=10/P1400:MC3(5),ECR3/IR60:BNQ
 /MC2(10),MMAS (500),BNQ(1)&JOSEPH : TDEN=137,TDEN=161,DURE=13,TLIM=400,COEF=20/I
 R360:MC1(66) ,DISQ1 /MC3(5) , ECR3
 &ARTHUR : TDEN=150,TDEN=170,DURE=110,TLIM=500,COEF=100/ PDP:MC4(30),IMP4/MRAP(90
)/MC2(10),MMAS (500),BNQ(1)
 &RENNARD:TDEN=152,TDEN=172,DURE=92,TLIM=520,COEF=200/IR60:MRAP(600),BNQ/
 MC3(5),PROG3(1) /IR360:MC1(45),LEC1
 &CRISTIAN: TDEN=155,TDEN=180,DURE=59,TLIM=270,COEF=12/IR360:MC1(40),DISQ1 8/

DEMANDES. ACCEPTEES: GASTON , ROBERT , ALPHA , PIERRE , DUPONT , MARTIN , LOULOU , ARTHUR , BERNARD , ,
DEMANDES REFUSEES: GAMMA , VICTOR , ROGER , OSCAR , JUJO , TOTO , PAUL , BOB , DURANT , JULES , JOEL ,

DATE EVENEMENTS Taux d occupation des ressources Satisfaction

MC1 *LEC1*DISQ*IMP1*MC2 *LECC*BNU2*DISQ*MC3 *DISQ*ECR3*TRAC*PRG*MC4 *IMP4*LECC*MRAP*BNQ *DISQ*MMAS*

T= 0	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	
------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	--

FIN DE LA SIMULATION: 9 DEMANDES ACCEPTEES 15 DEMANDES REFUSEES
VALEUR FINALE DE LA FONCTION DE SATISFACTION: 622

0001 EUS:RES02 ,13:17:56

0002 COS:RES13 ,13:17:56

#EXEC RES03:IMP4*SER:01(0011),ABORT=(NOALFON)

DEMANDES	ACCEPTÉES	GASTON	ROBERT	GAMMA	ROGER	(METHODE OPTIMISEE)	PIERRE	JUJU	DUPONT	PAUL	LOULOU	JULES	DUPANLOU	ARTHUR
DEMANDES	REFUSEES:	ALPHA	VICTOR	OSCAR	TOTO	MARIN	BOB	BORIS	DURANT	JOEL	JOSEPH	CHRISTIAN		
DATE	EVENEMENTS	TAUX D OCCUPATION DES RESSOURCES												
		SATISFACTION												

TAUX D'OCCUPATION DES RESSOURCES

SATISFACTION

$$MC1 * LEC1 * DISG * IMP1 * MC2 * LEC2 * BND2 * OISW * MC3 * ISU * FCR4 * TRAC * PRIG * NCA * IMPA * LEC * MPAD * NND * ISUMMAS +$$
[illegible]

FIN DE LA SIMULATION:

	13 DEMANDES ACCEPTEES	11 DEMANDES REFUSEES
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	1	1
21	1	1
22	1	1
23	1	1
24	1	1
25	1	1
26	1	1
27	1	1
28	1	1
29	1	1
30	1	1
31	1	1
32	1	1
33	1	1
34	1	1
35	1	1
36	1	1
37	1	1
38	1	1
39	1	1
40	1	1
41	1	1
42	1	1
43	1	1
44	1	1
45	1	1
46	1	1
47	1	1
48	1	1
49	1	1
50	1	1
51	1	1
52	1	1
53	1	1
54	1	1
55	1	1
56	1	1
57	1	1
58	1	1
59	1	1
60	1	1
61	1	1
62	1	1
63	1	1
64	1	1
65	1	1
66	1	1
67	1	1
68	1	1
69	1	1
70	1	1
71	1	1
72	1	1
73	1	1
74	1	1
75	1	1
76	1	1
77	1	1
78	1	1
79	1	1
80	1	1
81	1	1
82	1	1
83	1	1
84	1	1
85	1	1
86	1	1
87	1	1
88	1	1
89	1	1
90	1	1
91	1	1
92	1	1
93	1	1
94	1	1
95	1	1
96	1	1
97	1	1
98	1	1
99	1	1
100	1	1

0000, 15:15:15, 15:10:40

7 JS

E(J) : 1018, C.A.T.H.R.: 12⁰⁰72, 13:18:40

[illegible]


```

RESS / IR360:MC1(128),LEC1,DISQ1(5),IMP1(2)/IR50 :MC2 (256),LECC2,BND2(7),
DISQ2(6)/PI1400 : MC3(250),DISQ3(4),ECR3(3),TRAC3(2),PRUG3(50) / PDP : MC4(64),
IMP4,LECC4 / IR60 : MRAP(1000),BNQ(4) / ICL : DISQ6(6),MMAS(512)
*REQU&RENE :TDEM=100,TDEB=110,DURE=90,TLIM=160,COEF=6 /IR360:MC1(80),DISQ1(3),IMP
1/IR50:MC2(183),LECC2,BND2(4) /PI1400:MC3(100),DISQ3(2),ECR3(5),PRUG3(40)
/PDP:IMP4,LECC4,IR60:MRAP(500)/ ICL:DISQ6(4)
&CRISTIAN : TDEM=155,TDEB=180,DURE=59,TLIM=270,COEF=12/IR360:MC1(40),DISQ1
&DANIEL : TDEM=158,TDEB=300,DURE=22,TLIM=400,COEF=20 /IR60:MRAP(100)/MC2(90),BND2
/PDP:MC4(24),LECC4
&ERNEST : TDEM=170,TDEB=200,DURE=25,TLIM=400,COEF=28/PI1400:PRUG3(10),ECR3,IRAC3
/IR360 :MC1(20),DISQ1 / IR50:MC2(104),DISQ2(2)
&FERNAND : TDEM=172,TDEB=195,DURE=70,TLIM=450,COEF=20/PDP:MC4(10),IMP4/IR60:
MRAP(100)
&GEORGES : TDEM=178,TDEB=195,DURE=70,TLIM=450,COEF=50/ICL:MMAS(300),DISQ6/IR360:
MC1(16),IMP1
&HANS : TDEM=180,TDEB=220,DURE=32,TLIM=400,COEF=15/IR360:IMP1,MC1(40),DISQ1 /
BND2(4),DISQ2/ICL: MMAS(256),DISQ6
&SIDORE : TDEM=180,TDEB=200,DURE=150,TLIM=400,COEF=115/IR 50 :MC2(128),
BND2(4),DISQ2/ICL: MMAS(256),DISQ6
&JEANPAUL : TDEM=185,TDEB=250,DURE=32,TLIM=400,COEF=210/IR360:MC1(64),DISQ1/
IR 50 :MC2(86),LECC2(1),BND2(2)
& KAFKA :TDEM=185,TDEB=200,DURE=62,TLIM=500,COEF=44 / ICL:DISQ6(3),MMAS(200)/
PI1400:MC3(10),PRUG3 /IR60:MRAP(403),BNQ
&LAURENT :TDEM=180,TDEB=210,DURE=10,TLIM=400,COEF=5/PI1400:ECR3,IRAC3,MC3(20)
/PDP:MC4(24),LECC4
&MICHEL : TDEM=186,TDEB=205,DURE=105,TLIM=600,COEF=110 /IR360:MC1(64),IMP1,DISQ1
/ICL : DISQ6,MMAS(300)
&NICOLAS : TDEM=187,TDEB=250,DURE=92,TLIM=550,COEF=120 / IR50:MC2(30),
DISQ2 / IR60:MRAP(1000),BNQ(1)
/IR50:MC2(25),BND2
&OSCAR : TDEM=189,TDEB=235,DURE=15,TLIM=545,COEF=5/IR60:MRAP(100),BNQ/
PI1400:MC3(20),PRUG3
/ICL : DISQ6,MMAS(300)
&PAULETTE : TDEM=189,TDEB=280,DURE=22,TLIM=600,COEF=10/IR360:MC1(32),DISQ1
/IR50:MC2(25),BND2
&GULTURE : TDEM=192,TDEB=222,DURE=69,TLIM=599,COEF=70 / PDP:MC4(16),LECC4
/ICL: MMAS(512),DISQ6(2)
&KUGER : TDEM=195,TDEB=233,DURE=96,TLIM=491,COEF=155/PI1400:MC3(64),DISQ3,ECR3,
PRUG3(10)/IR60:MRAP(600),BNQ
/ICL : DISQ6,MMAS(300)
/IR50:MC2(25),BND2
&SERGE :TDEM=195,TDEB=212,DURE=128,TLIM=428,COEF=61/IR360:MC1(64)/ICL:MMAS(99)
&THEODULE : TDEM=198,TDEB=290,DURE=49,TLIM=550,COEF=50/ PI1400:MC3(200),PRUG3(10)
/MRAP(10)
/ICL : DISQ6,MMAS(300)
&URSULE : TDEM=200,TDEB=240,DURE=145,TLIM=600,COEF=190/IR360:IMP1,MC1(10)/
PI1400 :MC3(69),ECR3/IR60:MRAP(400),BNQ
&VICTOR : TDEM=201,TDEB=231,DURE=65,TLIM=555,COEF=101 /ICL:DISQ6,MMAS(100)
/PDP:MC4(24),LECC4
&WILLY : TDEM=205,TDEB=240,DURE=32,TLIM=200,COEF=95 / IR360:MC1(128)/ICL:MMAS
(150)/IR50 :MC2(30),DISQ2
/PDP:MC4(24),LECC4
&XAVIER :TDEM=206,TDEB=230,DURE=71,TLIM=400,COEF=98/IR360:MC1(64)/PI1400:MC3(10),
TRAC3,PRUG3(13)&
YVES:TDEM=207,TDEB=220,DURE=70,TLIM=350,COEF=105/IR360:MC1(64),LEC1/IR50:MC2(99)
&
ZOE:TDEM=207,TDEB=240,DURE=89,TLIM=400,COEF=185/IR60:MRAP(100),BNQ/PI1400:MC3(20)
&GASTON :TDEM=210,TDEB=230,DURE=50,TLIM=600,COEF=14/ IR360: MC1(12)/
IR50 :MC2(11)/PI1400 : PRUG3(5)/ IR60: MRAP(2) &

```


DEMANDES ACCEPTEES: CRISTIAN, ERNEST , FERNAND , ISIDORE , JEANPAUL, LAURENTI , MICHEL , NICOLAS , OSCAR , PAULETTE, JULIURE , ROGER ,									
DEMANDES REFUSEES: RENE , DANIEL , GEORGES , HANS , KAFKA , THEODOULE, WILLY , ZOE , GASTON ,									
DATE EVENEMENTS Taux d Occupation des Ressources									
SAISFACIION									
T=0	0*	0*	0*	0*	0*	0*	0*	0*	0*
T=180	40*	0*	0*	0*	0*	0*	0*	0*	0*
T=195	40*	0*	0*	0*	0*	0*	0*	0*	0*
T=200	60*	0*	0*	0*	0*	0*	0*	0*	0*
T=210	60*	0*	0*	0*	0*	0*	0*	0*	0*
T=220	60*	0*	0*	0*	0*	0*	0*	0*	0*
T=225	40*	0*	0*	0*	0*	0*	0*	0*	0*
T=230	104*	0*	0*	0*	0*	0*	0*	0*	0*
T=239	104*	0*	0*	0*	0*	0*	0*	0*	0*
T=250	64*	0*	0*	0*	0*	0*	0*	0*	0*
T=265	128*	0*	0*	0*	0*	0*	0*	0*	0*
T=282	128*	0*	0*	0*	0*	0*	0*	0*	0*
T=290	128*	0*	0*	0*	0*	0*	0*	0*	0*
T=301	76*	0*	0*	0*	0*	0*	0*	0*	0*
T=329	76*	0*	0*	0*	0*	0*	0*	0*	0*
T=350	76*	0*	0*	0*	0*	0*	0*	0*	0*
T=351	128*	0*	0*	0*	0*	0*	0*	0*	0*
T=352	106*	0*	0*	0*	0*	0*	0*	0*	0*
T=374	74*	0*	0*	0*	0*	0*	0*	0*	0*
T=446	74*	0*	0*	0*	0*	0*	0*	0*	0*
T=461	74*	0*	0*	0*	0*	0*	0*	0*	0*
T=479	74*	0*	0*	0*	0*	0*	0*	0*	0*
T=497	64*	0*	0*	0*	0*	0*	0*	0*	0*
T=584	0*	0*	0*	0*	0*	0*	0*	0*	0*
T=589	0*	0*	0*	0*	0*	0*	0*	0*	0*
T=653	0*	0*	0*	0*	0*	0*	0*	0*	0*

FIN DE LA SIMULATION:

VALEUR FINALE DE LA FONCTION DE SAISFACIION:

19 DEMANDES ACCEPTEES

1614

7 DEMANDES REFUSEES

0002 EOS:RES03 ,17:35:58

0003 BOS:RES02 ,17:35:58

#EXEC RES02:KAMRAN:D1(D00110),ABURI=(NOACTION)

RESS/IBM360:DS(1),DISK(2)/P1100:BM/ ORU:D,ES/
ALPHA:A(12),A1234567890,IPM/
P1200:IMPR(5)*REQU &PAU:IDEM=1,IDE8=2,DURE=3,TLIM=55, COEF=1/IBM360:DS(1)/D
/ALPHA:A(1)/IPM,8M,BM/ES8
JEAN:IDEM=4,IDE8=6,DURE=50,TLIM=20,COEF=20/US,DISK,BM
&MARCEL:IDEM=5,IDE8=20,DURE=12, TLIM=40,COEF=20/A(3),IMPR ,D,ES
PIERRE:IDEM=10,IDE8=20,DURE=30,TLIM=55,COEF=100/ IBM360:DISK/A1234567890123
456789,IPM/D
&D4:IDEM=10,IDE8=20,DURE=30,TLIM=30,COEF=50/US,DISK(2),BM,D,ES,A(6),IPM,IMPR(4)&
JACQUES:IDEM=11,IDE8=20,DURE=30,TLIM=55,COEF=100/ IBM360:DISK/A1234567890123
&ALFRED:IDEM=50,IDE8=80,DURE=30,TLIM=100,COEF=50/BM,D,IMPR(2),A(5),A123456/,IPM
&ALBERT:IDEM=50,IDE8=80,DURE=30,TLIM=100,COEF=50/BM,D,IMPR(2),A(1),A123456/,IPM
,ES&/

DEMANDES ACCEPTEES:PAU ,JEAN ,MARCEL ,JACQUES ,ALFRED ,
DEMANDES REFUSEES: PIERRE ,D4 ,ALBERT ,

DATE EVENEMENTS Taux d occupation des ressources Satisfac tion

Taux d occupation des ressources									
US	*DISK*BM	*D	*ES	*A	*A123*IPM	*IMPR*			
T= 0	*	*	*	*	*4567*	*	*	*	*
T= 2DEBUT DE PAU	*	0*	0*	0*	0*	0*	0*	*	1*
T= 5FIN DE PAU	*	1*	1*	1*	1*	1*	1*	*	21*
T= 6DEBUT DE JEAN	*	0*	0*	0*	0*	0*	0*	*	41*
T= 20DEBUT DE MARCEL	*	1*	1*	1*	1*	1*	1*	*	141*
T= 32DEBUT DE JACQUES	*	2*	2*	2*	2*	2*	2*	*	191*
T= 50FIN DE MARCEL	*	1*	1*	1*	1*	1*	1*	*	*
T= 56FIN DE JEAN	*	1*	1*	1*	1*	1*	1*	*	*
T= 80DEBUT DE ALFRED	*	0*	0*	0*	0*	0*	0*	*	*
T= 110FIN DE ALFRED	*	0*	0*	0*	0*	0*	0*	*	*

FIN DE LA SIMULATION: 5 DEMANDES ACCEPTEES 3 DEMANDES REFUSEES
VALEUR FINALE DE LA FONCTION DE SATISFACTION: 191

0003 EOS:RES02 ,17:36:17

0004 BOS:RES03 ,17:36:17

#EXEC RES03:NAME\$=D1(D00110),ABUR1=(NOACTION)

REFERENCES

- 1 - HABERMAN, A.N. Prévention of System Deadlocks
Comm. of the ACM 12, 7 (july 1969), 373-377,385.
- 2 - HOLT, Richard C. Comments on Prevention of System Deadlocks
Comm. of the ACM 14, 1 (Jan. 1971), 36-38)
- 3 - Coffman Jr, E.G. System Deadlocks.
ACM Computing Surveys 3, 2 (June 1971), 67-68
- 4 - DENNING, P.J. Third Generation Computer Systems
Computing Surveys 3, 4 (Dec. 1971), 175-216
- 5 - HAVENDER, J.W. Avoiding deadlock in multitasking systems.
IBM Systems Journal 7, 2(1968), 74-84
- 6 - MAHL R. An Analytical Approach to computer Systems Scheduling (Ph.D. Dissertation). University of Utah, Computer Science department, June 1970.
- 7 - HOLT, R.C. Deadlock in computer systems. Ph. D ; Thesis, Rep. TR-71-91, Cornell Univ. , Dept of Computer Science, Ithaca, N.Y., 1971.
- 8 - Rapport d'analyse du projet CYCLADES ; L. POUZIN, Juin 1972
- 9 - WILKINSON, P.T. A model for the local area of a data communications network- Software organisation. ACM Data communications Symposium, Pine Mountains, Oct. 1969.
- 10 - DAVIES, D.W. The principles of a Data Communication Network for Computers and Remote Peripherals, I.F.I.P., 1968
- 11 - ROBERT, L.G. and WESSLER, B.D. Computer Network Development to Achieve Resource Sharing. AFIPS Conference Proceedings May 1970.
- 12 - KLEINROCK, L., Analytic and Simulation Methods in Computer Network Design. AFIPS Conference Proceedings May 1970.
- 13 - ELIE, M., General Purpose Networks of Computers ; Masters Thesis.; Dept. of Computer Science Universite de Californie à Los Angeles (UCLA) 1970.
- 14 - CARR, S. CROCKER, S. and Cerf, V., Host-Host Communication Protocol in the ARPA Network, AFIPS Conference Proceedings May 1970.

Vu.

es membres du jury :

MM. MALECOT
KOULOUMDJIAN
MAHL

Vu et approuvé.

Le Directeur de l'U.E.R.
de MATHEMATIQUES

J. BRACONNIER

Vu et accordé le permis d'imprimer.

Lyon, le 6 Octobre 1972

Le Président de l'Université
Claude Bernard,

J. BOIDIN

